Adapter and Integration Guide

Operations Center 5.5

November 18, 2014

Legal Notices

Contents

About This Guide

The *Adapter and Integration Guide* explains how adapters enable the Operations Center server to connect to and communicate with third-party management systems.

Audience

This guide is intended for Operations Center system administrators.

Feedback

We want to hear your comments and suggestions about this manual and the other documentation included with this product. Please use the *User Comments* feature at the bottom of each page of the online documentation.

Additional Documentation & Video Tutorials

This guide is part of the Operations Center documentation set. For the most recent version of the *Adapter and Integration Guide* and a complete list of publications supporting Operations Center, visit our Online Documentation Web Site at Operations Center 5.5 online documentation.

The Operations Center documentation set is also available as PDF files on the installation CD or ISO; and is delivered as part of the online help accessible from multiple locations in Operations Center depending on the product component.

Also watch our instructional videos on select topics from the *Tutorials for NetIQ Operations Center* playlist on the NetIQ YouTube page.

Additional Resources

We encourage you to use the following additional resources on the Web:

- NetIQ User Community: A Web-based community with a variety of discussion topics.
- NetIQ Support Knowledgebase: A collection of in-depth technical articles.
- NetIQ Support Forums: A Web location where product users can discuss NetIQ product functionality and advice with other product users.

Technical Support

You can learn more about the policies and procedures of NetIQ Technical Support by accessing its Technical Support Guide.

Use these resources for support specific to Operations Center:

- Telephone in Canada and the United States: 1-800-858-4000
- Telephone outside the United States: 1-801-861-4000
- E-mail: support@netiq.com
- Submit a Service Request

Documentation Conventions

A greater-than symbol (>) is used to separate actions within a step and items in a cross-reference path. The > symbol is also used to connect consecutive links in an element tree structure where you can either click a plus symbol (+) or double-click the elements to expand them.

When a single path name can be written with a backslash for some platforms or a forward slash for other platforms, the path name is presented with a forward slash to preserve case considerations in the UNIX* or Linux* operating systems.

A trademark symbol (®, ™, etc.) denotes a NetIQ trademark. An asterisk (*) denotes a third-party trademark.

1 Overview

Adapters enable the NetIQ Operations Center server to connect to and communicate with third-party management systems.

The Operations Center platform ships with adapter support for many third-party products. However, when an out-of-the-box is not available or does not meet your specific needs, Operations Center offers various tools to customize your integration with nearly any application, database, or data source.

The following sections provide an overview of the different ways to integrate to third-party products and tools:

1.1 Adapter Integrations to Third-Party Products

Adapters are configured with a one-to-one relationship, where one adapter talks to a specific management system, discovery tool, or trouble ticket system. An adapter is Operations Center’ interface with supported third-party products.

Adapters are written to the management system’s API when feasible; most, but not all, integrate bidirectionally with the management system. In some cases, the adapter communicates directly with the management system. In other cases, communication occurs through an Object Request Broker (ORB) supplied by Operations Center.

The basic steps for integrating to third-party products are:

1. Additional integration and configuration steps might be necessary to integrate some network and systems management products, discovery tools, and trouble ticket systems. Some adapters require installation of ORB software.

   For instructions per product, see:

   - Chapter 3, “Application and Management System Integrations,” on page 23
   - Chapter 4, “Discovery Tool Integrations,” on page 143
   - Chapter 5, “Trouble Ticket Systems Integrations,” on page 165

   For information on ORBs, see Chapter 10, “ORB Installation,” on page 249.
2. Create and configure an adapter for each integration.

   For instructions on creating adapter instances, see Section 2.1, “Creating an Adapter,” on page 17.

For high severity issues, we recommend that you contact Support.

1.2 Custom Integrations Using Operations Center Integration Tools

Additional Operations Center tools allow integration with third-party products when an adapter integration is not available or does not meet your specific needs.

For example, if may want to integrated with:

- A home-grown help desk system.
- An application that pings devices on a routine basis.
- A database using a REST interface to integrate the data.

In many cases, the data you may want to integrate is not always management data, it can include customer details, application details, etc.

Table 1-1 outlines the available options to pull data into Operations Center when it can't be done with a standard Operations Center third-party adapter.

Table 1-1 *Additional Integration Tools*

Integrating data from...	Use...	Description
Databases	Data Integrator	Provides an IDE (Integrated Development Environment) that an administrator uses to define an integration directly into a database, which includes the definition of: ◆ Topology, a way to create elements. For example, devices. ◆ Element Properties. For example, IP Address, MAC Address. ◆ Alarms ◆ Relationships. For example, server1.netiq.com communicates via HTTP to server2.netiq.com. ◆ Mechanisms to reference external performance data. Data Integration definitions, once built and deployed, can be used on any Operations Center server. For information on using the F/X Adapter, see *Operations Center 5.5 Data Integrator Guide*.
Log Files, ASCII Streams, SOAP	Event Manager	Monitors log files, receive ASCII streams, SOAP and several other options. The raw data is processed using predefined rule sets and the output is surfaced in Operations Center as events and alarms. Alarms can consist of standard alarm messages that provide the status of network components, messages that create or delete elements, or messages that initiate certain actions. Noise events are dropped or correlated and previously opened alarms are automatically closed. For information on using the Event Manager, see *Operations Center 5.5 Event Manager Guide*.

Integrating data from...	Use...	Description
Web Sites and Web Applications	Experience Manager	Conducts end user, synthetic testing on applications and Web sites and measures performance. Identify and resolve potential infrastructure issues before customers experience problems. Experience Manager emulates end-user business processes against applications on a 24x7 basis, including Web and non-Web environments, and applications. For information on using the Event Manager, see *Operations Center 5.5 Experience Manager Guide*.
XML Files	F/X Adapter	Translates XML files from third party products. The administrator updates configuration files to instruct the adapter how to use the data to create objects/elements/ devices as well as alarms. For information on using the F/X Adapter, see *Operations Center 5.5 F/X Adapter Guide*.
Pre-formatted Text Stream from TCP Port	Universal Adapter (formerly the Script Adapter)	Listens on a TCP port and process a pre-formatted text stream into elements/alarms. The administrator either writes a Java script that is launched when the adapter is started; or writes an external script or application (using perl, java, C, etc) to integrate with and pull data from the third party product. For information on using the Universal Adapter, see Chapter 7, “NOC Universal Adapter,” on page 211.
SNMP Polling	SNMP Integrator	Polls specific MIB values on a routine basis to understand health and availability. The process starts by importing a MIB, setting up a topology (how the data is organized under the adapter) and then setting up the polling (IE: what MIB value to poll, how often, define good/bad results). There are some “discovery” capabilities, but this is not a full fledge discovery tool. It is ideal for filling some monitoring gaps (IE: polling a VPN device for active sessions or an application for exceptions/errors) that are not easily filled with other existing third party products due to political or technical barriers. For information on using the F/X Adapter, see *Operations Center 5.5 SNMP Integrator Guide*.

In addition, an integration might:

- Open a ticket when a service goes critical/red using Automation Events.

  For information about automations, see “Defining and Managing Automation Events” in the *Operations Center 5.5 Server Configuration Guide*.

- Launch the native console via right-click action on an element.

  For more information about setting up console connections, see Chapter 8, “Establishing Console Connections,” on page 219.

  For more information about custom right-click options, see “Modifying Element and Alarm Menus”.

- Integrate report information into the Dashboard using iFrames.

  For information about automations, see "Integrating Reports from External Reporting Tools Using iFrames" in the *Operations Center 5.5 Dashboard Guide*.

1.3 Demonstration Adapters

The following adapters available in Operations Center are used only for demonstration purposes and are not documented:

- Business Metric Demo
- NOC Demonstration Adapter

2 Creating Adapters

An adapter is the Operations Center interface with the management systems in a network. Operations Center cannot interact with any management system without adapters.

When adapters exist for a management system, Operations Center's auto-discovery capability locates all the objects in a connected network and displays them in the browser. It is important to define adapters for each instance of a management system in the network. For example, if IBM Tivoli NetView runs on three servers, define an Operations Center adapter for each of the three instances of NetView.

Use an InterConnection Adapter to view remote servers and allow users to build distributed views, or enable server communications through a firewall.

The following sections cover creating and maintaining adapters:

2.1 Creating an Adapter

Additional integration and configuration steps are necessary to integrate some network and systems management products, discovery tools, and trouble ticket systems before you create an adapter.

To watch a video tutorial for creating adapters, go to:


http://www.youtube.com/watch?v=aS7U8OD9ofE

To create an adapter:

1 Verify any required configuration tasks have been performed for the specific integration.

For specific product sections, see:

- Chapter 3, "Application and Management System Integrations," on page 23
- Chapter 4, "Discovery Tool Integrations," on page 143
- Chapter 5, "Trouble Ticket Systems Integrations," on page 165

2 In the *Explorer* pane, expand *Administration* > *Adapters*,.

3 Right-click *Adapters*, then click *Create Adapter* to open the Create Adapter dialog box.

4 Select the *Type* of adapter and customize the default name to one that describes the management system, such as `NetView TestLab`.

5 Configure settings in the Properties table as needed.

5a For information about the properties for the adapter type, see Chapter A, "Adapter Property Reference," on page 279.

5b Scroll through the values in the properties table and locate the string `replace with hostname where … was installed` in the *Value* column, then enter the name of the host on which the management server or ORB runs.

5c Continue scrolling through the adapter properties table and replace every instance of the above string, as well as the string `replace with name of …` with the correct name.

6 To automatically start the adapter after starting the Operations Center server, select *Start adapter automatically*. Deselect this option to start the adapter manually after starting the Operations Center server.

7 Click *Create*.

The new adapter displays in the *Explorer* pane after it starts.

If two adapters share the same name, the adapter created last has "(1)" appended to the end of its name.

2.2 Starting, Stopping, or Deleting an Adapter

Adapters can be started or stopped individually or as a group. Use caution before deleting any adapter as there might be settings you wish to save first in order to recreate the adapter later.

The following sections provide steps for starting and stopping adapters:

- Section 2.2.1, "Starting, Stopping, or Deleting an Adapter," on page 18
- Section 2.2.2, "Starting or Stopping All Adapters," on page 19

2.2.1 Starting, Stopping, or Deleting an Adapter

IMPORTANT: Deleting an adapter deletes all regular expressions used to match elements associated with the adapter. These expressions can be viewed in the *Match* tab on the Elements property page for a Service Model element. Before deleting an adapter, you might want to copy these regular expressions, if it is necessary to re-create them later.

To start, stop or delete an adapter:

1 In the *Explorer* pane, expand *Administration* > *Adapters*.

2 Right-click an adapter, then click one: *Start Adapter*, *Stop Adapter*, or *Delete Adapter*.

The adapter starts or stops.

If you select *Delete Adapter*, when the confirmation dialog box displays, click *Yes*.

2.2.2 Starting or Stopping All Adapters

To start or stop all adapters:

1. In the *Explorer* pane, expand *Administration*.
2. Right-click the *Adapters* element, then click *Start all adapters* or *Stop all adapters*.

2.3 Modifying Adapter Properties

Adapter properties vary by adapter type. For information on specific adapter properties, see Appendix A, "Adapter Property Reference," on page 279.

To modify an adapter by changing its property values:

1. In the *Explorer* pane, expand *Administration* > *Adapters*.
2. Right-click an adapter, then click *Properties* to open the Status property page.
3. In the left pane, click *Adapter* to display the adapter properties table.
4. Specify changes in the *Value* column.
5. Click *Apply* and close the property pages.

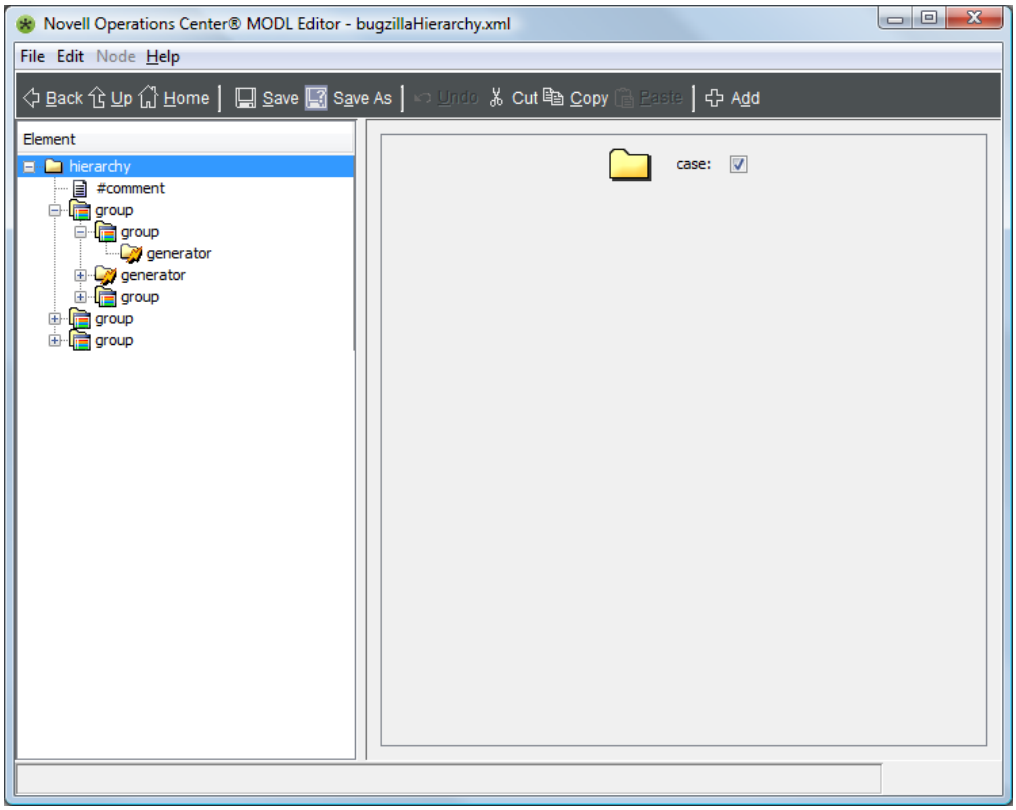
2.4 Customizing the Adapter Hierarchy

Many adapters can use an adapter hierarchy file, which can be modified using the Operations Center XML Editor. For example, for a BMC Remedy ARS adapter, set up the adapter hierarchy to group trouble tickets by schemas. Then within a schema, group tickets by assignment and status.

To edit the adapter hierarchy file:

1. In the *Explorer* pane, expand *Administration* > *Adapters*.
2. Right-click an adapter, then click *Edit Hierarchy Definition*.

   The Operations Center MODL Editor opens in a new window:



3 Update the hierarchy file as required.

For more information about using the Operations Center MODL Editor for editing XML, see "The Operations Center XML Editor" in the *Operations Center 5.5 Server Configuration Guide*.

For more information about hierarchy files, see Chapter 9, "Using the HierarchyFile," on page 227.

2.5 Using a NOC InterConnection Adapter (ICA)

The main purpose of creating an InterConnection Adapter is to view remote servers and allow users to build distributed views.

Environments that contain multiple management systems, databases, firewalls and other management components have different options for the overall configuration. An InterConnection Adapter (ICA) enables server communications through firewalls.

The InterConnection adapter connection mechanism operates in the same way as the Operations Center console-to-server connection.

It is possible to use the ICA over an HTTPS port. The ICA follows a process of determining the other required ports. If the default of `CORBA.bidir=true` has not changed in the `/OperationsCenter_install_path/html/classes/CORBA.properties` file, the server uses one port for bidirectional IIOP communications.

The following sections detail using an InterConnection adapter:

- Section 2.5.1, "Creating an InterConnection Adapter," on page 21
- Section 2.5.2, "State Propagation and Expected Behavior with the InterConnection Adapter," on page 21

2.5.1 Creating an InterConnection Adapter

To create an InterConnection Adapter:

1. Create an adapter for *NOC - InterConnection Adapter*.

   For general instructions, see Section 2.1, "Creating an Adapter," on page 17.

2. Define the adapter properties.

   When specifying InterConnection adapter properties (see Table A-32, "Operations Center InterConnection Adapter Properties," on page 338), specify the host name and HTTP port of other Operations Center server. For example, enter `Mosserver:8080`, or just enter `Mosserver` because 8080 is the default HTTP port.

2.5.2 State Propagation and Expected Behavior with the InterConnection Adapter

The state of the InterConnection Adapter and its elements is not reported to the Operations Center server and because of this, does NOT contribute to the state of the *Administration* and *Elements* nodes. InterConnection Adapter element states only propagate to the Service Views (*Layout* view) for *Services* hierarchy elements.

In addition, many features are NOT available for InterConnection Adapter elements in the *Services* hierarchy. For example:

- Creation of new element operations.
- Pre-existing class-specific operations.
- *Elements* and *Condition* property pages.
- Alarm filters.

3 Application and Management System Integrations

Operations Center can communicate with the application and management systems listed in this section.

For information about supported versions of a specific application or management system, see the *Operations Center 5.5 Getting Started Guide*.

For the most recent information concerning a specific adapter or ORB, see the *Operations Center 5.5 Release Notes*.

The following sections describe configuration steps for each application and management system integration:

3.1 Amazon Elastic Compute Cloud (Amazon EC2)

Amazon Elastic Compute Cloud (Amazon EC2) is a Web service that provides resizable compute capacity in the cloud. It allows you to upload VM images and run them within the host provider.

Operations Center integrates with Amazon EC2 to allow you to monitor and manage the following objects:

- Amazon Machine Images (AMIs)
- Instances
- Elastic block volumes
- Snapshots
- Elastic IPs

Use the Amazon EC2 adapter to:

- View the relationships between Amazon EC2 objects. For example, all instances launched from a particular AMI show the instances as children under the AMI.
- Click on any instance to view the AMI from where it was launched.
- Launch on demand instances and attach EBS volumes to instances.
- Monitor an instance’s CPU utilization and other performance metrics from the Operations Center console. Note that monitoring must be turned on.

NOTE: Launching instances and starting instances from the Amazon EC2 adapter incurs charges against your AWS account.

Operations Center supports the launching of the following Amazon EC2 instance types: Large (m1.large), Extra Large(x1.xlarge), High-Memory Extra Large (m2.xlarge), High-Memory Double Extra Large (m2.2xlarge), High-Memory Quadruple Extra Large (m2.4xlarge), and High-CPU Extra Large (c1.xlarge).

To integrate Amazon EC2:

1. Create an Amazon Elastic Compute Cloud adapter.

   For instructions, see Section 2.1, “Creating an Adapter,” on page 17.
2. Modify the adapter properties.

   For instructions, see Section A.1, “Amazon Elastic Compute Cloud (Amazon EC2),” on page 280.

3.2 Blade Logic Operations Manager

The Blade Logic adapter interfaces with a Blade Logic Application server and a variety of management applications on that server to gather XML data and process them into the Managed Objects Normalized Data format (MOND). MOND is similar to MODL (Managed Objects Definition Language), which is an XML format for defining hierarchies. MOND defines an XML format for interpreting events, alarms, relationships, and performance data.

Refer to the following topics to integrate with Blade Logic Operations Manager:

- Section 3.2.1, "Integrating Blade Logic Operations Manager," on page 25
- Section 3.2.2, "Configuring the Blade Logic Operations Manager Monitor," on page 26
- Section 3.2.3, "Managed Objects Normalized Data (MOND) Format DTD," on page 28

3.2.1 Integrating Blade Logic Operations Manager

To integrate Blade Logic:

1. Create a Blade Login Operations Manager adapter for each instance of Blade Logic on the network.

   For instructions, see Section 2.1, "Creating an Adapter," on page 17.

2. Modify the adapter properties.

   For instructions, see Section A.2, "Blade Logic Operations Manager," on page 281.

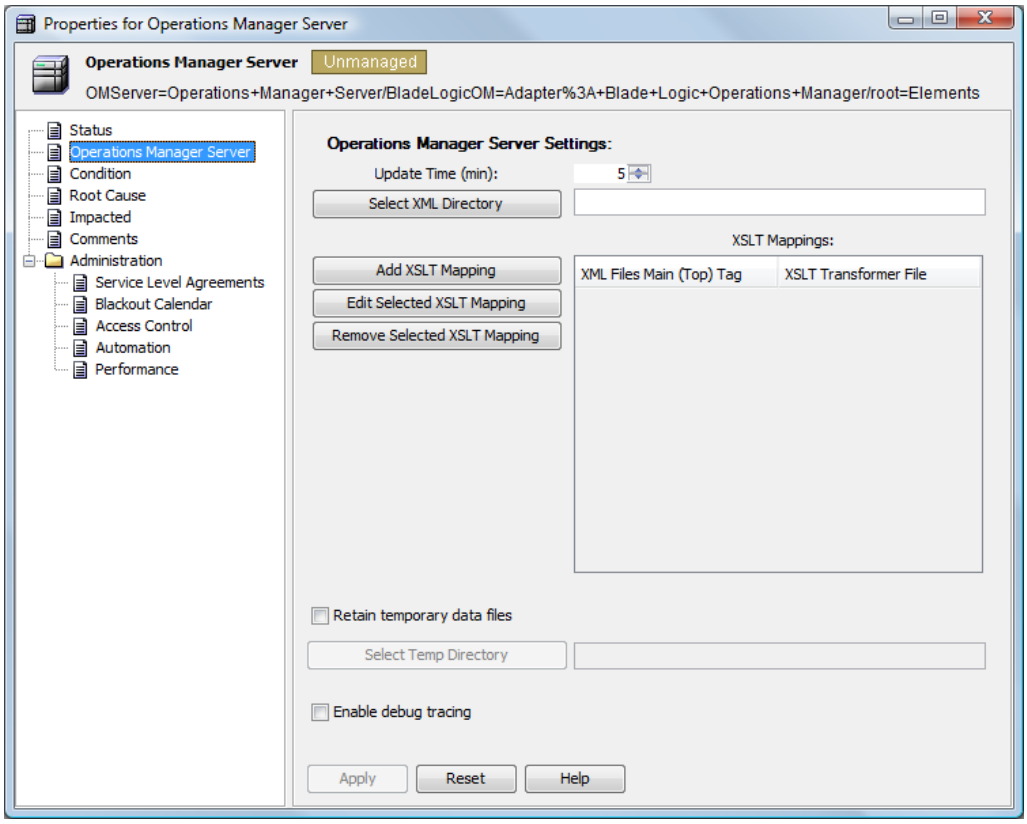
3. Configure the Blade Logic Operations Manager Monitor.

   For instructions, continue to Section 3.2.2, "Configuring the Blade Logic Operations Manager Monitor," on page 26.

3.2.2 Configuring the Blade Logic Operations Manager Monitor

To configure the Blade Logic Operations Manager Monitor:

1. In the *Explorer* pane, click *Elements* to expand it, then click the Blade Logic Operations Manager adapter to expand it.
2. Right-click the *Operations Manager Server*, then click *Properties*.
3. In the left pane, click *Operations Manager Server*.



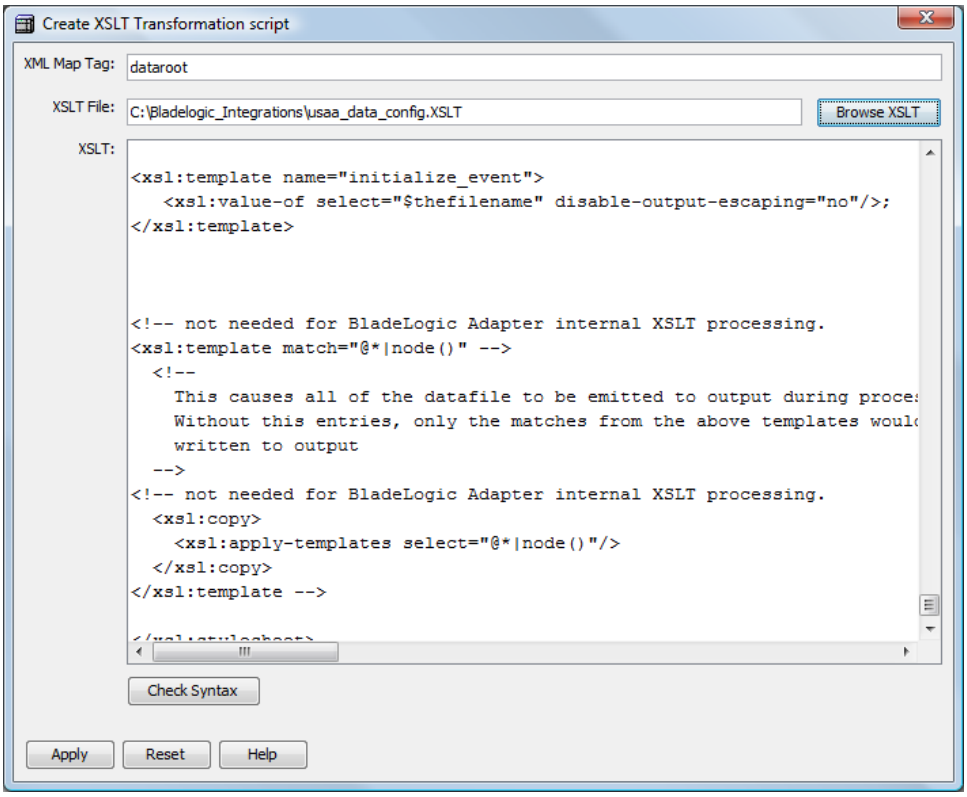
4. In the *Update Time* field, to check the XML directory for new XML files, specify the time interval in milliseconds.
5. Click *Select XML Directory* to browse and select a directory for storing the XML files.

   The Monitor collects all XML files from the selected directory.
6. Click *Add XSLT Mapping* to create XSLT to XML root tag mapping to open the XSLT mapping editor, then do the following:

   6a Create the XSLT directly in the editing panel, or click *Browse* and select an existing XSLT, which you can edit and/or modify.

   Enter as many mappings as you want the Monitor to format into the Managed Objects Normalized Data format (MOND) (see Section 3.2.3, "Managed Objects Normalized Data (MOND) Format DTD," on page 28).

The following illustrates an XSLT file in the editor:



6b Add the root tag of the XML files that this XSLT will translate.

The XML root tag is the first content tag in an XML file. The following is example code:

```
<?xml version="1.0" encoding="UTF-8"?>
   <dataroot xmlns:od="urn:schemas-microsoft-com:officedata"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:noNamespaceSchemaLocation="CI_Data.xsd" generated="2005-02-
    22T15:19:00">
   <CI_Data>
       <ID>1</ID>
       <Application>Enigma Corp</Application>
       <CI>mdkse</CI>
       <IT_Service>ABC</IT_Service>
       <Domain>AIX Web</Domain>
       <Location>BKDC2,RED2,FEDSA</Location>
   </CI_Data>
   <CI_Data>
      <ID>2</ID>
      <Application>Enigma Corp</Application>
      <CI> mdkse </CI>
      <IT_Service>ABC</IT_Service>
      <Domain>AIX Web</Domain>
      <Location> BKDC2,RED2,FEDSA </Location>
   </CI_Data> . . .
```

7 To save files read from the XML directory into a directory, select *Retain temporary data files*. Then, click *Select Temp Directory* to select the directory for storing temporary data files.

8 To enable debug tracing in the `formula.trc` file, select *Enable debug tracing*.

9 Click *Apply*.

10 In the *Explorer* pane, right-click the *Operations Manager Server* element, then select *Manage* > *Start Monitoring* to begin pushing XML data into the adapter.

3.2.3 Managed Objects Normalized Data (MOND) Format DTD

The following code is the DTD for the formatting of MOND output by XSLT translators for the Blade Logic adapter:

```
<?xml version="1.0" encoding="ISO-8859-1"?>

<!--
The XML file can contain a "events" element that
surrounds one or more event objects.
-->
<!ELEMENT events (event)+ >

<!--
The event chunk contains name<->value pairs either as "name=value"
or as an xml <field name="field name">field value</field>.
The class is the element class.
The isalarm is "false" if only elements are to be generated, if it
is set to "true" then alarms are created as well.
The severity denotes the severity of alarms if they are to be created
The state is the state of the alarm if alarms are being created
-->
<!ELEMENT event (#PCDATA|field)* >
<!ATTLIST event
          class CDATA #REQUIRED
          isalarm (true|false) "false"
          severity (unknown|ignore|ok|info|minor|major|critical) "ok"
          state CDATA #IMPLIED>

<!ELEMENT field (#PCDATA) >
<!ATTLIST field
          name CDATA #REQUIRED>

<!--
The performance element contains "measures" which are tagged to
elements.
-->
<!ELEMENT performance (measure)+ >

<!--
measures contain an attribute to the element that it applies to
and one or more values.
-->
<!ELEMENT measure (value)+ >
<!ATTLIST measure
          elementName CDATA #REQUIRED
          measureName CDATA #REQUIRED>
<!--
values have a numerical value and a time attribute to give the actual
measured value
-->
<!ELEMENT value (#PCDATA) >
<!ATTLIST value
          timevalue CDATA #REQUIRED>


<!--
The relationship element has links that create relationship links
```

```
between 1 or more elements
-->
<!ELEMENT relationship (link)+ >

<!--
a link has 2 endpoints and a type, endpoints are references to elements
-->
<!ELEMENT link (endpoint1,endpoint2) >
<!ATTLIST link
          linktype CDATA #REQUIRED>

<!ELEMENT endpoint1 (#PCDATA) >

<!ELEMENT endpoint2 (#PCDATA) >
```

3.3 BMC Software Event Manager

Use the BMC Software Event Manager to integrate to BMC ProctiveNet Performance Management (BPPM), formerly SMC Event Manager.

To integrate Event Manager:

1. Create a BMC Event Manager adapter.

   For general instructions on creating adapters, see Section 2.1, "Creating an Adapter," on page 17.

2. Accept the defaults for the adapter properties and click *Create*.

   In general, it is acceptable to use these defaults. For more information about adapter properties, see Section A.4, "BMC Software Event Manager," on page 283Section A.4, "BMC Software Event Manager," on page 283.

   For detailed descriptions of the adapter properties, see Section A.4, "BMC Software Event Manager," on page 283.

3. Start the adapter.

4. Do the following to configure the adapter:

   4a Right-click the adapter instance under *Elements*, and select *Properties.*

   4b In Properties, open *Event Manager Administration*.

   4c In the *General* tab, specify the following:

   - **Web Services Host URL:** Address of the IIWS server. Specify the path portion of the URL if it is different from `/imws/services/ImpactManager`. For example: `http://164.99.17.72:9003` or `http://164.99.17.72:9003/imws/services/ImpactManager`.
   - **Polling Filter:** Name of a filter defined on the IIWS server that determines the alarms received after initial discovery. Once the Web Services Host URL is populated, the Browse feature (magnifying glass icon) is available to load polling filters from the IIWS server.
   - **Polling Interval (seconds):** How often the adapter polls for events after initial discovery.
   - **BAROC Query:** The query in BAROC syntax (a BMC proprietary syntax) that identifies the events loaded during initial discovery.

Replace ***ADD IIWS CELL NAME HERE*** with the cell name of the IIWS server (usually `iiwsgatewayserver`). The default query retrieves all non-closed events that normally propagate to the IIWS server. If the string `{EVENT-CLASS}` is included in the query, the query is executed once for each valued specified in *{EVENT-CLASS} Substitution Values*.

- **{EVENT-CLASS} Substitution Values:** List of event classes to discover. Event classes in the BPPM environment are hierarchical and `EVENT` is typically the root of that hierarchy. All sub-classes of a requested event class are included, so there is no need to specify another event class is the root type is specified. Add, edit or delete substitution values.

4d In the *Cells* tab, add cell configurations to be discovered. The adapter only receives updates from these cells after initial discovery if they are configured to propagate events to the IIWS server. Propagation rules are configured in the BPPM environment. Click the Add icon to add the cell configurations to be discovered, then specify:

- **Cell Name:** Name of the cell in the BPPM environment.
- **Passive discovery:** When selected, this cell is not queried during initial discovery in adapter start-up. If the adapter receives an event from the cell during it's polling process, then normal discovery of the cell is initiated.
- **Use Custom Delivery Query:** When selected, the enclosed *BAROC Query* and *{EVEN-CLASS} Substitution Values* settings are used instead of the defaults specified in the *General* tab.
- **Use Custom Credentials:** When selected, the *Username* and *Password* settings are used instead of the defaults specified in the *Advanced* tab.

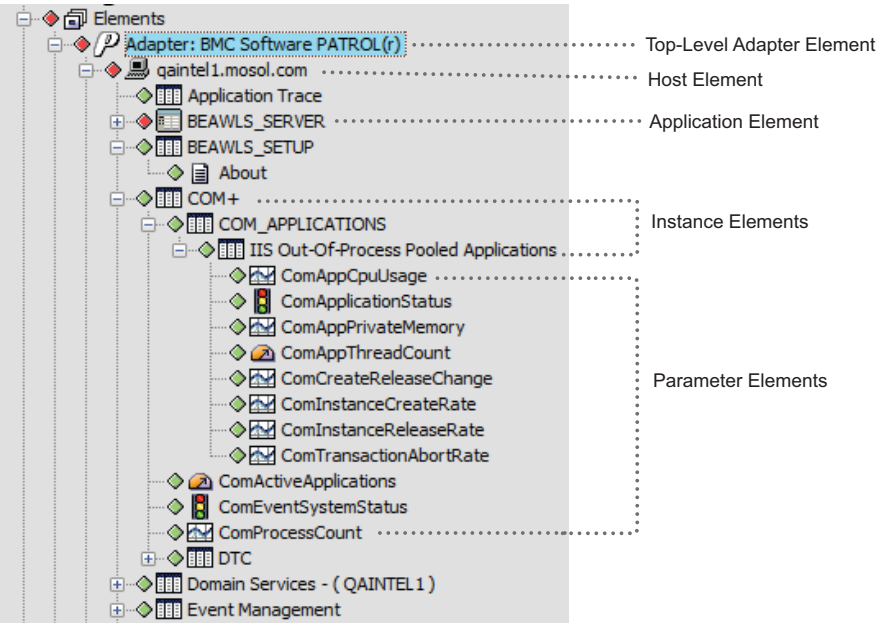
4e *(Optional)* If necessary, use settings in the Advanced tab to further configure the adapter:

- **Max Events Requested Per Fetch:** Maximum number of the events to retrieve per SOAP request. A larger value here can improve performance but may require more memory in the IIWS server process.
- **Communications Timeout (seconds):** Number of seconds to wait for a response from the IIWS server.
- **Max Retry Count (use 0 for unlimited):** Number of times to retry after a communications failure.
- **Retry Interval (seconds):** Number of seconds to wait between each retry.
- **Default Cell Credentials:** Username and Password used to communicate with cells (not typically required).

3.4 BMC Software PATROL

PATROL manages data regarding hosts, all applications discovered on these hosts, and parameters that are discovered for hosts and applications. The PATROL adapter integrates to PATROL to manage and display this data in Operations Center.

Figure 3-1 *PATROL hosts, applications, instances and parameters display under the PATROL adapter in the Explorer pane*



Operations Center displays all hosts and other objects defined in PATROL. The element hierarchy, as illustrated in Figure 3-1, displays as follows:

- The top-level element displays directly beneath Elements.
- Hosts display beneath the top-level element.
- Elements that represent the applications defined for a host display beneath each host.
- Each application has a container object (instance element) and contains parameters defined for the application.

You can recognize the type of object by looking at the element icon or class, as described in Table 3-1:

Table 3-1 *Reference showing how PATROL objects map to*

PATROL Object Type	Element Class	Element Icon Examples
Host	patrol_*AgentType* For example, `patrolNT` or `patrolLinux`	
Application	patrolAppl	
Instance	patrolInst	
Parameter	patrolParm	

The following sections provide instructions for integrating to PATROL:

- Section 3.4.1, “Integrating to PATROL,” on page 32
- Section 3.4.2, “Configuring the PATROL Integration,” on page 32
- Section 3.4.3, “Issuing Commands to Agents Using a System Output Window,” on page 41
- Section 3.4.4, “Performing Knowledge Module (KM) Commands,” on page 42

3.4.1 Integrating to PATROL

To integrate to PATROL:

1. If running on Solaris, enable Daemon Shell Access for shell access in the Operations Center Configuration Manager.

   For details, see "Daemon Pane" in the *Operations Center 5.5 Server Configuration Guide*.

2. Create a BMC Software PATROL adapter for each instance on the network.

   For instructions, see Section 2.1, "Creating an Adapter," on page 17.

3. Configure adapter properties.

   Configure the *Property Page Permissions* property to set the access rights for Hosts, Libraries, Communication Settings, and Host property pages. This controls a user's ability to view and edit the property pages of PATROL elements that display in the Operations Center console.

   For for a detailed list of properties, including the *Property Page Permissions* property, see Section A.5, "BMC Software PATROL," on page 284.

4. Add a library, one ore more hosts, and configure communication settings.

   For instructions, continue to "Configuring the PATROL Integration" on page 32.

3.4.2 Configuring the PATROL Integration

The following sections describe how to configure the PATROL integration:

Adding and Managing Libraries

The first step in PATROL adapter configuration is adding a library used by PATROL hosts. This is the same library specified in the PATROL_HOME environment variable in your existing PATROL implementation.

Operations Center supports multiple libraries to enable managing multiple agent versions. Using multiple library installations enables an administrator to manage a set of PATROL agents using multiple versions of PATROL from one Operations Center installation.

For example, if an infrastructure is migrating from PATROL version 7.0 to PATROL version 7.5, the administrator can create two library definitions, one for each version. Later, when creating the host/agent definitions for the adapter, select one of these library definitions, depending on which version of PATROL is running on the target agent. After upgrading an agent, the administrator simply changes the library used for the agent, then restarts the connection in Operations Center for that agent.

If upgrading from a previous version, for instructions on copying PATROL libraries to the Operations Center server, see the *Operations Center 5.5 Server Installation Guide*.

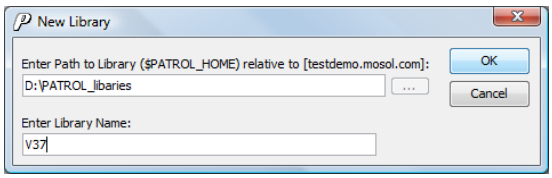
Adding a Library

To add a PATROL library:

1. In the *Explorer* pane, expand *Elements*.
2. Right-click the top-level BMC Software PATROL element, then click *Properties*.
3. In the left pane, click *Libraries*.
4. Click *New Library*.
5. In the New Library dialog box, specify the directory location of PATROL_HOME relative to the Operations Center server,

   or

   click the ... button and locate the directory, then specify a library name. The browse button is only available if you are running the Operations Center console local to the Operations Center server.



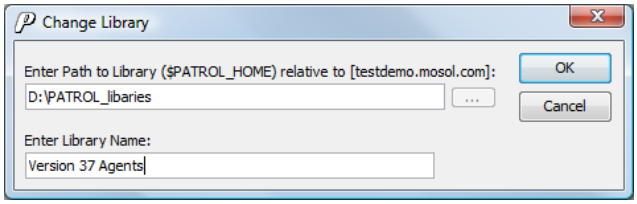
6. Specify a descriptive name for the library in the *Enter Library Name* field.

Changing a Library Location or Name

Change a library's location or name through the top-level PATROL element property pages.

To update a library location or name:

1. In the *Explorer* pane, expand *Elements*.
2. Right-click the top-level BMC Software PATROL element, then click *Properties*.
3. In the left pane, click *Libraries*.
4. Select the library name and then click *Change Library*.



5. Specify the new name and/or path for the library.

Changing a Library Assigned to a Host

To change a library assigned to a host:

1. In the *Explorer* pane, expand *Elements* > the top-level BMC Software PATROL element.
2. Right-click a host element, then click *Properties*.
3. In the left pane, click *Host*.
4. Select a new library from the *Knowledge Module Library* drop-down list.

Removing a Library

To remove a library:

1. In the *Explorer* pane, expand *Elements*.
2. Right-click the top-level BMC Software PATROL element, then click *Properties*.
3. In the left pane, click *Libraries*.
4. Select a library name and then click *Remove Library*.

Resolving Not Found Libraries

Removing a library that is assigned to one or more hosts has no visible effect during the current session. However, after stopping and restarting the adapter, sometimes a message states that Operations Center cannot find a library.

To resolve Libraries Not Found messages:

1. Check the libraries assigned to each host and find the hosts to which the deleted library is assigned.
2. Assign a different library to these hosts. For instructions, see "Changing a Library Assigned to a Host" on page 34.

Adding and Managing Hosts

In Operations Center, it is necessary to configure the hosts monitored by the PATROL integration. Configuring the PATROL adapter requires that you add one or more hosts to the PATROL adapter.

Hosts display beneath the top-level BMC Software PATROL element in the *Explorer* pane. On initial adapter startup, only the connected hosts display in the Hosts property page for the top-level BMC Software PATROL element.

Operations Center displays both monitored and unmonitored elements. However, Operations Center tracks and regularly updates conditions for monitored (or managed) hosts only. Unmonitored (or unmanaged) elements are displayed, but their condition is always UNMANAGED. This saves bandwidth and processor time, especially if a large section of network is designated as unmonitored /unmanaged.

The following topics explains how to add, monitor, and locate hosts; import configuration files, change host passwords; and remove hosts:

- "Changing Host Passwords" on page 37
- "Removing Hosts" on page 38

Adding Hosts

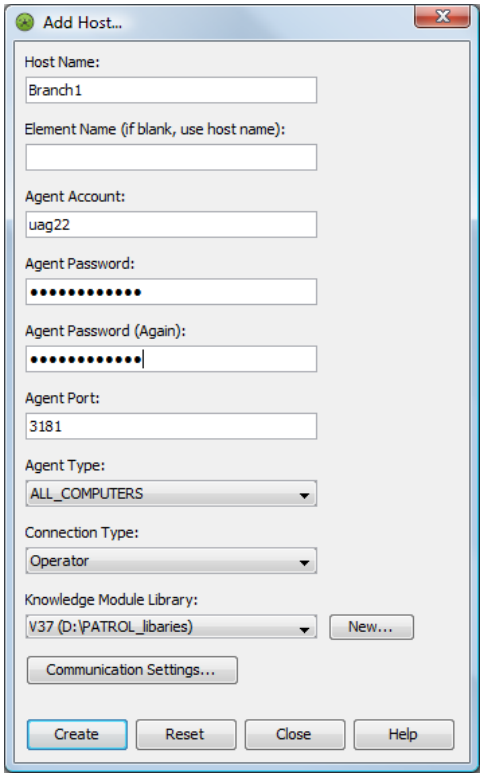
To add a host:

1 In the *Explorer* pane, expand *Elements*.

2 Right-click the top-level BMC Software PATROL element, do one of the following:

- Select *Add Host*.
- Select *Properties*. In the left pane, click *Hosts*, and then click *New Host*

The Add Host dialog box opens:



3 Fill in the host setting fields:

Host Name: The host name of the machine where the PATROL agent is running.

Element Name: The host definition of an element. If blank, the target host is used.

Agent Account, Password, Port, Type, Connection Type: For definitions of agent settings and the connection type, see your PATROL documentation.

Knowledge Module Library: Required. PATROL monitors and manages resources on hosts using information from files called Knowledge Modules (KMs). If multiple libraries exist, click the drop-down arrow to select a different library, then click *New* to define a new library.

For more information, see "Adding and Managing Libraries" on page 32.

4 To override the global settings and establish custom settings for an individual host, click *Communication Settings* on the Add Host dialog box and see "Establishing Communication Settings" on page 38 for additional information.

   Normally, the top-level BMC Software PATROL element sets communication settings globally for all hosts.

Monitoring Hosts

To monitor a host:

1 In the *Explorer* pane, expand *Elements* > the top-level BMC Software PATROL element.

2 To monitor a host, right-click a host element, then click *Manage* > *Start Monitoring*.

   The values in the *View* pane's *Condition* column change from *UNMANAGED* to a different value such as *CRITICAL* or *MAJOR*.

3 To stop monitoring a host, right-click a host element, then click *Manage* > *Stop Monitoring*.

   The unmanaged element and its subcomponents display in the *Explorer* and *View* panes, but Operations Center does not load their conditions. The values in the *Condition* column change to *UNMANAGED*.

Finding Hosts

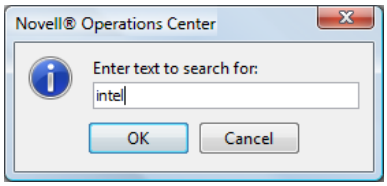
To locate a host in a long list on the Hosts property page:

1 In the *Explorer* pane, expand *Elements*.

2 Right-click the top-level BMC Software PATROL element, then click *Properties*.

3 In the left pane, click *Hosts* to open the Hosts property page.

4 Click *Find* to open the following dialog box:



5 Do one of the following:

- Specify a keyword that matches the host name.
- Use Perl 5 regular expressions to search for hosts. For example, `.*Acme.com` matches all host names containing Acme.com:

Importing Host Configuration Files

You can import PATROL host configuration files, `config.pet` files, and the PATROL Console `desktop.dt` file.

Because host passwords are imported, you can change them using the *Change Password* feature. For instructions see "Changing Host Passwords" on page 37.

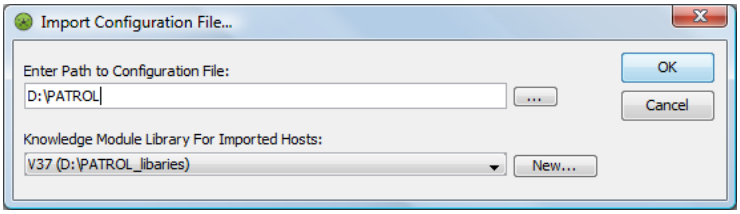
To import a host configuration file:

1 To create the PATROL `desktop.dt` file from your PATROL console, click *File*> *Desktop* > *Save*.

2 To import the configuration files, do the following from the Operations Center console:

2a In the *Explorer* pane, expand *Elements*.

2b Right-click the top-level BMC Software PATROL element, then click *Import Configuration File*.



2c Enter the path to the configuration file and select the Knowledge Module Library to use for imported hosts.

Changing Host Passwords

Entering an invalid password results in a normal connection, but the element is in the UNMANAGED state. A message states that an invalid password was used.

To change the password for an individual host:

1 In the *Explorer* pane, expand *Elements* > the top-level BMC Software PATROL element.

2 To change the password for an individual host:

2a Right-click a host element, then click *Properties*.

2b In the left pane, click *Host*.

2c Enter the new password in the *Agent Password* and *Agent Password (Again)* fields.

3 To change the password for one or more hosts through the top-level element:

3a Right-click the top-level BMC Software PATROL element, then click *Properties*.

3b In the left pane, click *Hosts*.

3c Select one or more hosts, then click *Change Password*.

Use `Ctrl` or `Shift` to select multiple host names.

3d Enter the new password, then click *Change*.

Removing Hosts

To remove a host using the element menu:

1. In the *Explorer* pane, expand *Elements* > the top-level BMC Software PATROL element.
2. Right-click a host, then click *Remove Host*.

Establishing Communication Settings

Communication settings establish basic connection properties between Operations Center and the PATROL agents, and enable the dynamic loading of applications that are not static.

Adding a new application through the PATROL Console results in displaying the new application in Operations Center, if the application loading interval is greater than zero. New applications do not display in Operations Center if the time out is less than zero.

Set communication settings at the global level for all hosts using the top-level PATROL element in the Operations Center *Explorer* pane, then override settings for a specific host.

The following topics describe configuring communication settings:

- "Identifying Static Applications in PATROL" on page 39
- "Configuring Global Communication Settings" on page 39
- "Saving Memory Resources on the Operations Center server" on page 40
- "Overriding Global Communication Settings" on page 40

Identifying Static Applications in PATROL

To identify static applications in PATROL:

1 Run the `PSL %DUMP KM_LIST` command, which lists *Yes* or *No* in the fourth column.

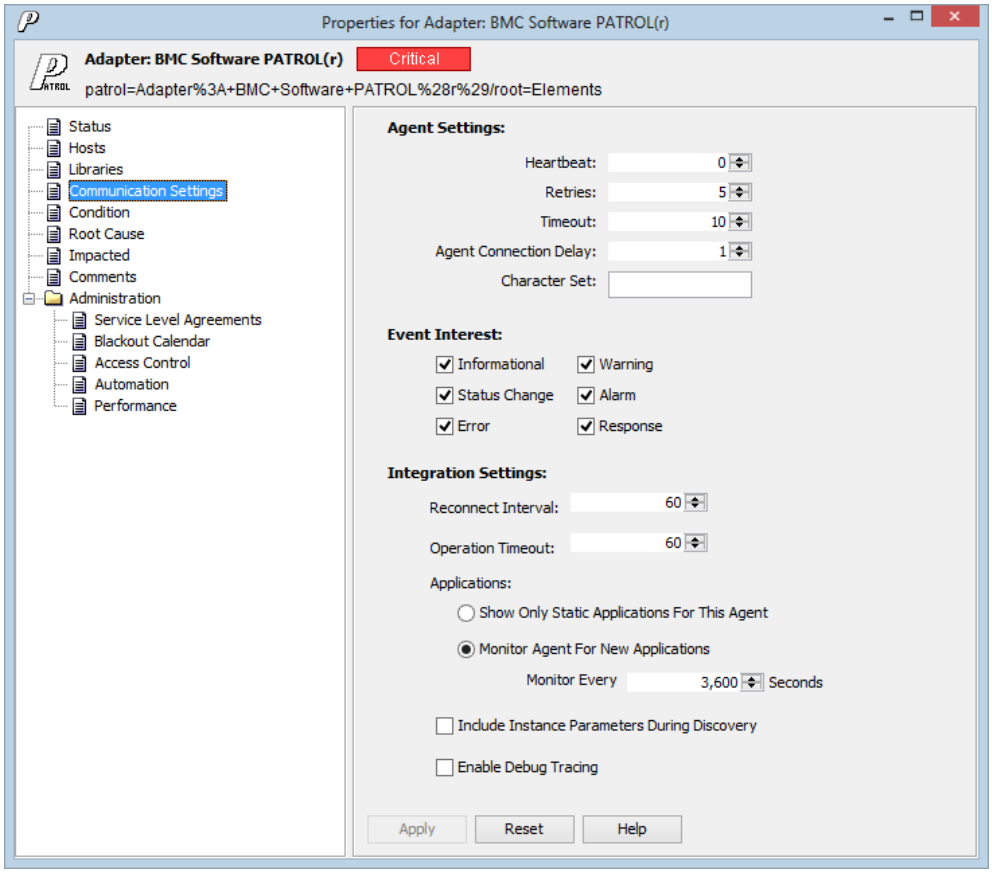
Configuring Global Communication Settings

To configure global communication settings:

1 In the *Explorer* pane, expand *Elements*.

2 Right-click the top-level BMC Software PATROL element, then click *Properties*.

3 In the left pane, click *Communication Settings*.



4 Specify *Agent Settings* to monitor the downstream connection between the agent and the client, which is the integration between PATROL and Operations Center:

Heartbeat: The amount of time between heartbeat packets that indicate the client is still communicating with the agent.

Retries: The maximum number of retry attempts.

Timeout: The maximum number of seconds to wait to establish a connection to the agent.

Character Set: A language standard for the encoding of characters from bytes. For example, BG18030 is the Chinese encoding standard.

5 Specify *Event Interest* settings by selecting the event types to monitor.

By default, all are enabled. Deselect event types that are not monitored.

6 Specify *Integration Settings*:

Reconnect Interval: The number of seconds between attempts to reconnect.

Operation Timeout: The number of seconds to wait before ceasing to send operation requests.

Show Only Static Applications for this Agent: Select to display current applications only and do not check for new applications.

Monitor Agent for New Applications: Select and specify the interval in seconds to check for new applications.

If new Knowledge Modules (KM) are loaded in the PATROL console, the new KMs appear in the agent tree and in Operations Center. Conversely, removing a KM causes an application to unload and Operations Center removes the object also.

Include Instance Parameters During Discovery: Leave unselected to discover PATROL parameter elements only as needed and save memory resources in Operations Center.

Because parameter elements contribute to the state of instances in Operations Center, they may have an impact your service views. When this is the case, or when parameters are included in the service views using dynamic matching, selecting this option allows those views to be correctly populated and updated on adapter start up.

WARNING: Selecting this option means a parameter discovery is performed during the initial connection to the agent, with the potential additional cost of time and memory.

---

7 Select *Enable Debug Tracing* only when instructed by Support for troubleshooting. When selected, a large amount of data is logged in the `formula.trc` file.

Saving Memory Resources on the Operations Center server

When considering the *Include Instance Parameters During Discovery* check box, note that leaving the option deselected can save memory resources in Operations Center. Instance parameter objects might not be important to provide value to a business view within Operations Center. However, instance parameters do help determine the state of instances. Many uses of the PATROL adapter include the presence of a particular instance in a business view, instead of a single parameter or parameters within a business view.

Furthermore, if using dynamic matching criteria in creating business views, deselecting the *Include Instance Parameters During Discovery* check box enables the Operations Center server to save memory while providing parameters to the business views.

To save memory resources on the Operations Center server:

1 Leave the *Include Instance Parameters During Discovery* option deselected in the Communication Settings property page.

Parameter data is not discovered until it is needed. Selecting this option means discovering the parameters during the initial connection to the agent, with the potential additional cost of time and memory.

Overriding Global Communication Settings

To override the global communication settings for a particular host and establish custom settings:

1 In the *Explorer* pane, expand *Elements* > the top-level BMC Software PATROL element.

2 Right-click a host element, then click *Properties*.

3 In the left pane, click *Host*.

4 Click *Communication Settings*.

5 Select *Use Custom Settings for this Agent*.

6 Specify the agent, event and integration settings for the host.

See Step 4 on page 39 in "Configuring Global Communication Settings" on page 39 for setting definitions.

3.4.3 Issuing Commands to Agents Using a System Output Window

Operations Center provides a terminal window console to issue Windows or UNIX commands, such as such as `%PSLPS`, to PATROL agents and get output.

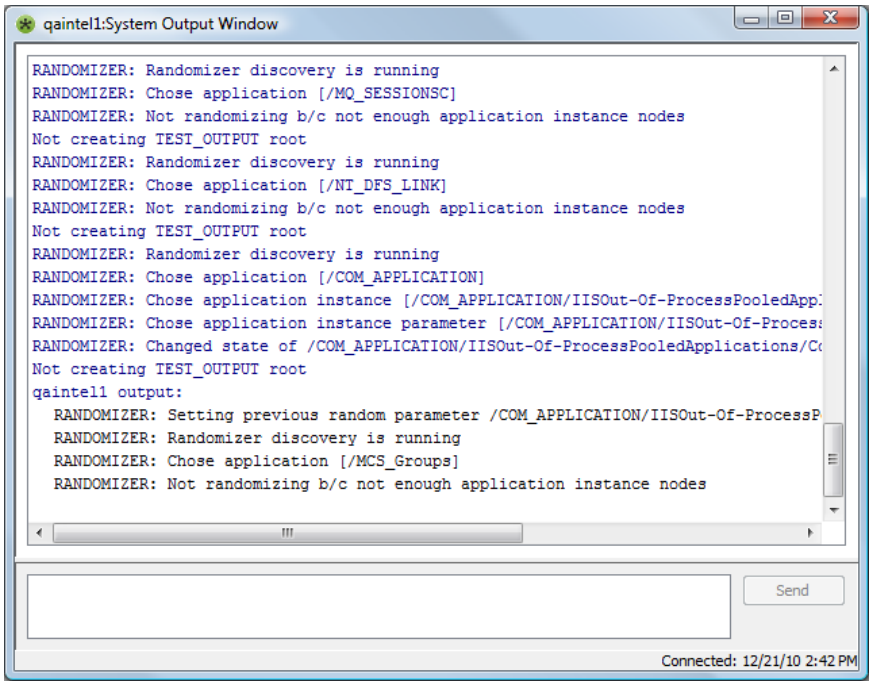
To interactively issue commands to agents:

1 To configure the System Output Window to show time stamps, set the *Show Timestamp in Agent System Output Window* to `true` in the adapter properties.

For information on PATROL adapter properties, see Section A.5, "BMC Software PATROL," on page 284.

2 In the *Explorer* pane, expand *Elements* > the top-level BMC Software PATROL element.

3 Right-click a host element, then click *Show Output Window* to open the System Output window:



4 In the System Output window, enter Windows or UNIX commands.

3.4.4 Performing Knowledge Module (KM) Commands

The KM commands that are available in the PATROL system surface in Operations Center through the KM Operations command.

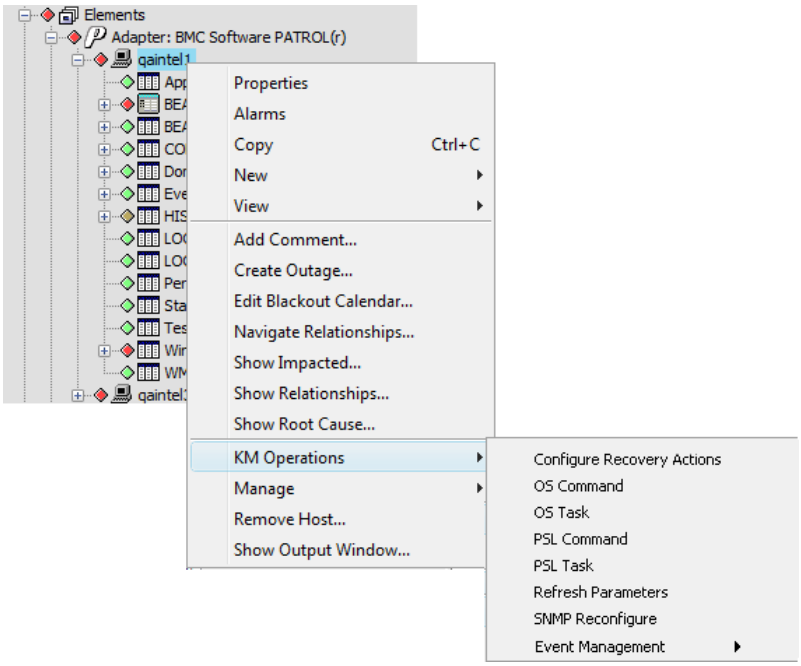
To access KM commands:

1 In the *Explorer* pane, expand *Elements* > the top-level BMC Software PATROL element.

2 Right-click a host element, click *KM Operations* and select a command.

Because the set of commands varies among customers, and users can create custom commands, Figure 3-2 shows only a sample of the available commands:

Figure 3-2 *PATROL Element Menu: This menu surfaces KM operations.*



3.4.5 Understanding PATROL Element Condition in Operations Center

As in PATROL, events (or alarms) do not affect the element conditions in Operations Center. The Operations Center adapter performs a status query to the PATROL application to retrieve the element's connection state.

IMPORTANT: Alarm severities have no influence on the state of PATROL adapter elements. For this reason, conditions may be counterintuitive as an element might be OK when it has many CRITICAL alarms.

An element condition change is any change that takes place on a resource that the PATROL application is monitoring. For example:

- A parameter goes above or below its normal range
- The connection state of a computer changes
- A parameter description changes for a class of objects

Table 3-2 lists the PATROL states and their corresponding element conditions.

Table 3-2 *Mapping PATROL Connection States to Element Conditions*

PATROL State	Operations Center Element Condition
VOID, UNKNOWN, INVALID	UNKNOWN
OFFLINE	INITIAL
OK	OK
WARN	MINOR
ALARM	CRITICAL

In order to register an instance condition change, Operations Center requeries all parameter element children for their updated status, since an *UpdParamState* event might not occur because of event filtering. Properties for a parameter element identify the attribute to monitor and the frequency to check it. These properties can include:

- A description of the monitored attribute
- The frequency for checking the attribute
- The instructions for measuring and monitoring the attribute
- Detection thresholds for abnormal attribute values

The condition of an instance element is then determined using a roll-up condition from parameter element children. For example, if the parameter element `Test > SWAP > Summary > SWAP > SwapTotSwapUsedPercent` changes to an ALARM state in PATROL, Operations Center sets its parent elements' conditions as follows:

1. `SwapTotSwapUsedPercent` changes to CRITICAL
2. `Summary` changes to CRITICAL
3. `SWAP` changes to CRITICAL
4. `Test` changes to CRITICAL

When the `SwapTotSwapUsedPercent` state changes to OK, all of its parent elements' conditions also change to OK.

A PATROL agent stores parameter commands and periodically executes them to retrieve data about the managed computer or application. Parameters add information to the built-in rules about computers and applications. Parameters also contain recovery actions that result from the parameter detecting an unhealthy state; for example, when a returned value falls outside specified alarm ranges.

3.4.6 Understanding PATROL Events in Operations Center

In the Operations Center *Alarms* view, new alarms display with parity to the PATROL event console. Standard alarm filtering occurs through the MODL (Managed Objects Definition Language™).

A PATROL event signifies the occurrence of a state change in a PATROL element. Examples of events include:

- Elevation of a parameter to an Alarm level
- Modification of a global parameter description
- Activation or deactivation of parameters
- Suspension or resumption of parameters
- Discovery of new applications

Table 3-3 lists PATROL event types and their corresponding Operations Center alarm severities.

Table 3-3 *PATROL Event Types and Corresponding Severity in Operations Center*

PATROL Event Type	Operations Center Alarm Severity	Filtered Out in Operations Center
ZERO, UNKNOWN	UNKNOWN	No
INFORMATION, RESPONSE	INFO	Yes
CHANGE_STATUS	OK	Yes

PATROL Event Type	Operations Center Alarm Severity	Filtered Out in Operations Center
WARNING	MINOR	No
ERROR	MAJOR	No
ALARM	CRITICAL	No

Each PATROL event also has a corresponding status: OPEN, ACKNOWLEDGED, CLOSED, ESCALATED, DELETED. By default, queries run only for OPEN, ACKNOWLEDGED and CLOSED events. State change events are disregarded as the latest state originates from the metadata discovery. You can filter events according to their status in the XML hierarchy file.

3.4.7 PATROL Element Conditions and Algorithms

To implement a persisted state, use the algorithm engine available via the Hierarchy file. For example, the following XML tag in the adapter hierarchy file causes elements to supply algorithms to control conditions:

```
<param name="usesAlgorithms" value="true" />
```

The algorithm engine must have available the following inputs:

- Current internal condition of the element in Operations Center
- Desired (or `True`) condition of the element in PATROL

Given these two items of information, it is possible to keep the condition at CRITICAL until it is explicitly set to OK. If the desired condition is not available, modify the PATROL adapter to provide the True condition of the remote object as input to the algorithm engine.

The algorithm can be a script. The previously documented `state` is now called `conditionState`, since state already has a meaning in all other NOC Script contexts. Also, a script can appear in the body of the `<exec command="script">` tag.

For more information, see Chapter 9, “Using the HierarchyFile,” on page 227.

3.5 BMC Software PATROL Enterprise Manager (PEM)

The following topics describe how to integrate with PATROL Enterprise Manager:

3.5.1 Integrating to PATROL Enterprise Manager

To integrate Patrol Enterprise Manager:

1. Create an adapter for each instance of PEM on the network.

   For instructions, see Section 2.1, “Creating an Adapter,” on page 17.
2. Modify the adapter properties.

   For instructions, see Section A.6, “BMC Software PATROL Enterprise Manager,” on page 285.

3.5.2 Integration Using a Secure Relay Connection

The PEM adapter can communicate with the PEM server using a relay connection to provide secure cross-host communications. The relay connection acts as an intermediary, accepting and delivering messages to one server to another.

All the relay connection components are installed automatically with the Operations Center product. Unzip the */OperationsCenter_install_path*/Relay.zip file and use the following instructions to set up the relay connection between Operations Center and the server:

- "Setting Up the Relay Connection Between Operations Center and the PEM Server" on page 46
- "Setting Security Parameters" on page 46

Setting Up the Relay Connection Between Operations Center and the PEM Server

To set up the relay connection between Operations center an the PEM server:

1. Follow the instructions for installing the relay application in the `readme.txt` file on the *Operations Center* CD.

   This includes information on modifying the `config/relay.properties` file.

2. Configure the PEM adapter properties that pertain to the relay.

   Note the following port requirements for using the relay connection:

   - The relay requires an open port for relay administration, set in the `mosrelay.adminPort` property in the */OperationsCenter_install_path*/config/relay.properties file. This port might not be used for any other purpose.
   - In addition, configure each relay with its own port for each required listener configuration. Refer to the */OperationsCenter_install_path*/config/relay.properties file for information on configuring relay listeners.

   Each relay is defined by a set of properties prefixed with `mosrelay.`*relay_name*, where *relay_name* is the name of the relay. Each relay definition must contain listener properties specifying how client connections are received. For required listener properties, see the */OperationsCenter_install_path*/config/relay.properties file.

Setting Security Parameters

The `mosrelay.`*relay_name*`.listener.security` property specifies the security level for connections accepted on the listener port. The three valid values are:

- **ssl:** All communications are encrypted with SSL.
- **sslWithClientAuth:** SSL with client certification authentication.
- **unsecured:** Clear text communications.

To support SSL, supply a trusted server certificate for the relay. If not using the default keystore in the `/config` directory, use the specified properties in relay.properties to point to the appropriate keystore.

Operations Center validates SSL certificate dates and flags certificates with expired dates or dates that are not yet valid.

If using self-signed certificates, the process for creating and trusting certificates is identical to the process for the Operations Center server.

If your relays are configured to verify client certificates, remember that the Operations Center server certificate must also be trusted by the VM running the relay.

On the adapter side, specify the following PEM adapter properties to transmit communications to/from the PEM server:

- **RelayServer:** Name of the server on which the relay connection exists.
- **RelayPort:** The port number configured for use by the PEM adapter for relay communications.
- **RelaySecurity:** The security level for the relay server: SSL or unsecured (meaning use clear text, which is not case-sensitive).

Note there are three valid values in the `config/relay.properties` file and two valid settings for the RelaySecurity adapter property. Important points to remember about these two settings: If the adapter is set to SSL security, the relay can be set to SSL or sslWithClientAuth. The names are not case sensitive.

In the case of sslWithClientAuth, the relay requests and validates the Operations Center server certificate as part of the SSL handshake. In either case, the adapter should be set to SSL.

3.5.3 Querying Historical Alarms

The *Query Alert History* right-click option on the PEM adapter element enables users to query historical alarms in the PEM persistent store. The *Query Alert History* option provides a pop-up window for users to specify start and stop date/time boundaries for displaying historical alarms. The historical alarms display for the length of time specified by the *AlarmAgeOutQueryExtDuration* and *AlarmAgeOutInSec* adapter properties.

For more information, see Chapter A, “Adapter Property Reference,” on page 279.

3.6 Castle Rock Computing SNMPc Network Manager

When Operations Center integrates to Castle Rock Computing SNMPc, the following occurs:

- SNMPc maps are imported into the Operations Center *Layout* view.
- SNMPc element hierarchies and alarms display in Operations Center as they display in the SNMPc client and optional MODL tree.

To integrate SNMPc:

1. Create an adapter for each instance of a SNMPc on the network.

   For instructions, see Section 2.1, “Creating an Adapter,” on page 17.

2. Modify the adapter properties.

   When there is a large SNMPc log file upon adapter startup, it can take an excessive amount of time to download all events in the log file. Use the *Alarms Discovery* property to filter the request for these events so that only pertinent information is downloaded through the adapter at startup. For example, filter by alarm condition to display everything except alarms with the OK condition, or filter by specifying the maximum age (in hours) of events to collect upon adapter startup.

   For property descriptions, see Section A.7, “Castle Rock Computing SNMPc,” on page 289.

3.7 Cisco Info Center (CIC)

To integrate Cisco Info Center:

1. Create an adapter for each instance of a CIC Info Server on the network.

   For instructions, see Section 2.1, “Creating an Adapter,” on page 17.
2. Modify the adapter properties.

   For property descriptions, see Section A.8, “Cisco Info Center,” on page 290.
3. If the CIC user account names do not match the Operations Center user account names, edit the /*OperationsCenter_install_path*/database/examples/cicaccountmap.properties file.

3.8 CiscoWorks2000 Device Fault Manager

Create an adapter for each instance of a CiscoWorks2000 DFM on the network (see Section 2.1, “Creating an Adapter,” on page 17). See the integration information in the following sections and then modify the adapter properties (see Section A.9, “CiscoWorks2000 DFM,” on page 293).

3.8.1 Integrating CiscoWorks2000 DFM

To integrate CiscoWorks2000 DFM:

1. Stop the Operations Center server.

   For instructions, see “Stopping the Operations Center server in Windows” and “Stopping the Server and the mosdaemon manually in UNIX” in the *Operations Center 5.5 Server Installation Guide*.
2. Configure the Operations Center server to not restart automatically.

   For more information, see the *Operations Center 5.5 Server Configuration Guide*.
3. Copy (or symbolically link) the `skclient.jar` file from the CiscoWorks distribution to the /*OperationsCenter_install_path*/classes/ext directory.

4 Update or install the license file through the Operations Center Configuration Manager, if applicable.

   The license file must contain one or more key entries for `com.mosol.integration.smarts.DfmIntegration`.

5 Restart the Operations Center server.

   For instructions, see "Manually Starting the Operations Center Server" and "Starting the Operations Center Server in UNIX" in the *Operations Center 5.5 Server Installation Guide*.

6 Create an adapter for each instance of CiscoWorks2000 DFM on the network.

   For instructions, see Section 2.1, "Creating an Adapter," on page 17.

   It is helpful to know the following information:

   - The hostname of the DFM Broker
   - The port that the DFM Broker listens on (default is 9002)
   - A valid user name and password, if DFM is running with authentication required

   IMPORTANT: The Operations Center server and CiscoWorks2000 DFM both run processes that use port 9002. If installing the Operations Center server and CiscoWorks2000 DFM on the same machine, change the Operations Center database listener port from 9002 to another port. Use the Operations Center Configuration Manager to change the Listener Port value from 9002 to an unused port number for the General Datastore and BSA and Alarm History values.

7 Modify the adapter properties.

   For property descriptions, see Section A.9, "CiscoWorks2000 DFM," on page 293.

3.8.2 Error Handling

The integration might throw exceptions in the case of errors or malfunctions. Operations Center logs all exceptions in a standard format. Each logging statement is associated with the following categories: ERROR, INFO, WARN and DEBUG.

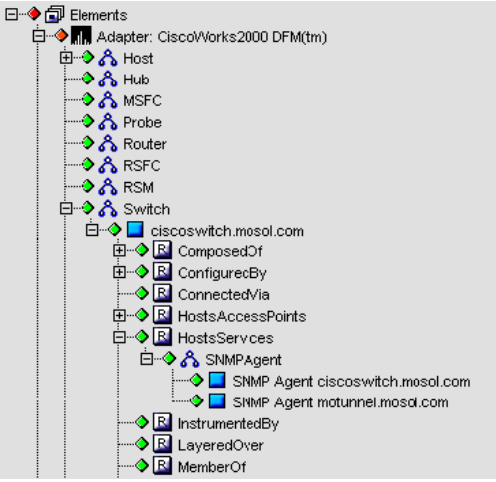
3.8.3 Understanding the Discovery of CiscoWorks2000 DFM Elements

The CiscoWorks2000 DFM adapter automatically performs element discovery and synchronization that include:

- Discovery of all the technology elements from the DFM database and populating the Operations Center object model
- Discovery of classes, instances, and relations between objects for use in the Service Views
- Relationship mapping between object for *n*-deep nested class/instance/relation sets
- Subscriptions to event notifications
- Management of instances (when available)
- Element property mapping and property pages for elements
- Automatic adapter updates for DFM instance create and delete notifications

Figure 3-3 shows CiscoWorks2000 DFM elements displaying under the adapter:

Figure 3-3 *Explorer Pane*



The specific objects are implemented as proxy objects of the actual DFM objects. When Operations Center queries values (or other data), the adapter requests the data on-demand from the DFM server. The adapter caches some common data values for efficiency.

Operations Center aggregates all data from the various consoles into the element hierarchy. When comparing the Operations Center element hierarchy tree to the DFM console, typically more Relation instances display in Operations Center than in a single DFM console. This is because the DFM console only shows the relations applicable to that console.

In Operations Center, a unique key value identifies each element. The key value consists of a concatenation of up to three string values delimited by a colon (:). This key value uniquely identifies the element instance in the Operations Center element repository.

A single element can have multiple parents in multiple locations in the element hierarchy. The element's key value consists of the concatenation of the root hierarchy key plus all element keys for each of its parents.

Table 3-4 describes DFM elements names.

Table 3-4 *DFM Element Names*

Element Type	Name in Operations Center consists of…
Class	The class name. For example: `Host`
Instance	The concatenation of the class with the instance name. For example: `Host:server.mosol.com`
Relation	The concatenation of the class, instance, and relation name. For example: `Host:server.mosol.com:ComposedOf`

3.8.4 Understanding Element Conditions

Normally, the default condition algorithm determines the condition of a Operations Center element. The default condition algorithm sets the element condition equal to the most critical severity of any alarm attached to the element or to its children. The exception is for relation elements, which do not propagate up their condition.

For information on changing the default algorithm, see Using Algorithms to Calculate Element State in the *Operations Center 5.5 Server Configuration Guide*.

3.8.5 Understanding Element Operations Permissions

The ability to perform operations on elements using the right-click menu depends on the ACL permissions granted. Table 3-5 outlines the ACL permission on various element menu operations.

Table 3-5 *CiscoWorks2000 DFM—ACL Permissions on Element Operations*

Element	Operation	ACL Permission
Adapter (top-level)	Add Subscription	Define
	Correlate Now	Manage
	Instance Create	Define
	Instance Delete	Define
	Recompute Codebook	Manage
	Repository Save	Define
	Repository Restore	Define
Instance	Manage	Manage
	Unmanage	Manage
	Delete	Define

3.8.6 Viewing Attributes, Details, Programs, and Libraries

The CiscoWorks2000 DFM adapter property pages provide a way to review information regarding attributes, programs, libraries and other features.

To view the adapter property pages:

1. In the *Explorer* pane, expand *Elements* > *Adapter:CiscoWorks2000*.
2. Right-click an element, then click *Properties* to open the Status property page.
3. Select different property pages to view specialized information about the Adapter, Instance, and Class elements.

Table 3-6 outlines the various information and features found in the property pages for the adapter root, class and instance elements.

Table 3-6 *CiscoWorks2000 DFM Adapter and Element Property Pages*

Element Type	Property Page	Description
Adapter Root	Details	Provides information about the adapter connection with the CiscoWorks Domain Manager, including status, socket port and timeout settings.
	Correlation	Provides information regarding the Correlation Codebook and its associated settings. For more information, see Section 3.8.13, "Correlating Events," on page 61.
	Programs	Displays the list of programs loaded in the server.
	Libraries	Displays the list of models loaded in the server.
	Subscriptions	Displays a list of event notification subscriptions (see Section 3.8.12, "Subscribing to Events," on page 59). Allows adding, changing, or deleting subscriptions.
Class	Description	Displays a textual description for the class element.
	Attributes	Provides the values of various attributes associated with the element.
	Events	Displays a list of simple events defined for this class. The simple event names display. Information on this page is read-only.
	Operations	Displays a list of all operations defined for this class.

Element Type	Property Page	Description
Instance	Attributes	Provides the values of various attributes defined for the class. *Refresh* updates the attribute values in real time. Attributes Access types: **MR_NO_ACCESS:** Attribute value is unavailable. **MR_STORED:** Sets the attribute value when the class is modeled or instantiated. If the attribute is not read-only, modify it at runtime (optional). **MR_COMPUTED:** Computes the attribute value upon request. Computed access types are always read-only. **MR_INSTRUMENTED:** Retrieves attribute value from an external source through a protocol. **MR_PROPAGATED:** Derives attribute's value from the values of other class instances in which this class is involved in a relationship or relationship set. **MR_UNCOMPUTABLE:** Cannot compute attribute value. **MR_COMPUTED_WITH_EXPRESSION:** Computes attribute value upon request, using an expression. Computed access types are always read-only.
	Events	Displays a list of simple events defined for this class. Allows subscribing and unsubscribing to event notifications for the instance element (see Section 3.8.10, "Understanding Event Notifications and Alarms Mappings," on page 56).
	Operations	Displays a list of all available operations for this element. Allows running a selected operation (see Section 3.8.8, "Running Instance Operations," on page 55).

3.8.7 Creating and Managing Instances

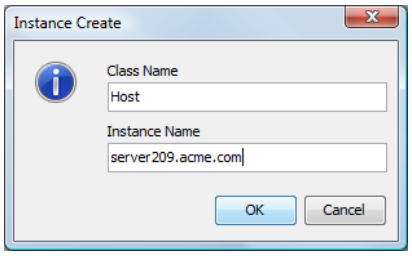
The DFM monitors managed instances. The DFM does not monitor unmanaged elements, but does probe them and stores element information in the DFM inventory.

When adding elements to or deleting elements from the Domain Manager from another client, the adapter receives notification about the event. The adapter automatically creates (or deletes) the appropriate class and instance elements, allowing for a lazy discovery of object relations.

Creating an Instance

To create an instance in the DFM:

1. In the Operations Center *Explorer* pane, expand the *Elements* > CiscoWorks2000 DFM element.
2. Right-click the CiscoWorks2000 DFM element, then click *Instance Create*.



3. Specify the class and instance names.

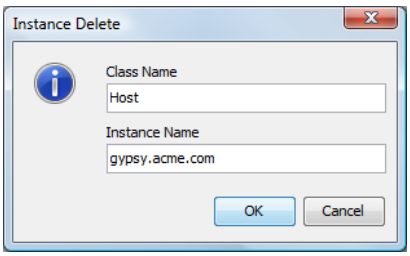
Deleting an Instance

When you delete an instance, you are deleting the instance and its children.

To delete an instance:

1. Do one of the following:
   - Right-click the instance element, then select *Delete*.

     Click *Yes* when prompted for confirmation.
   - In the Operations Center *Explorer* pane, right-click the *Elements* > CiscoWorks2000 DFM element and select *Instance Delete*.



   Specify the name of the class and instance.

Managing an Instance

To start managing an instance:

1. In the *Explorer* pane, right-click an instance element under the CiscoWorks2000 DFM element, then click *Manage*.

Unmanaging an Instance

To unmanage an instance:

1. In the *Explorer* pane, right-click an instance element under the CiscoWorks2000 DFM element, then click *Unmanage*.

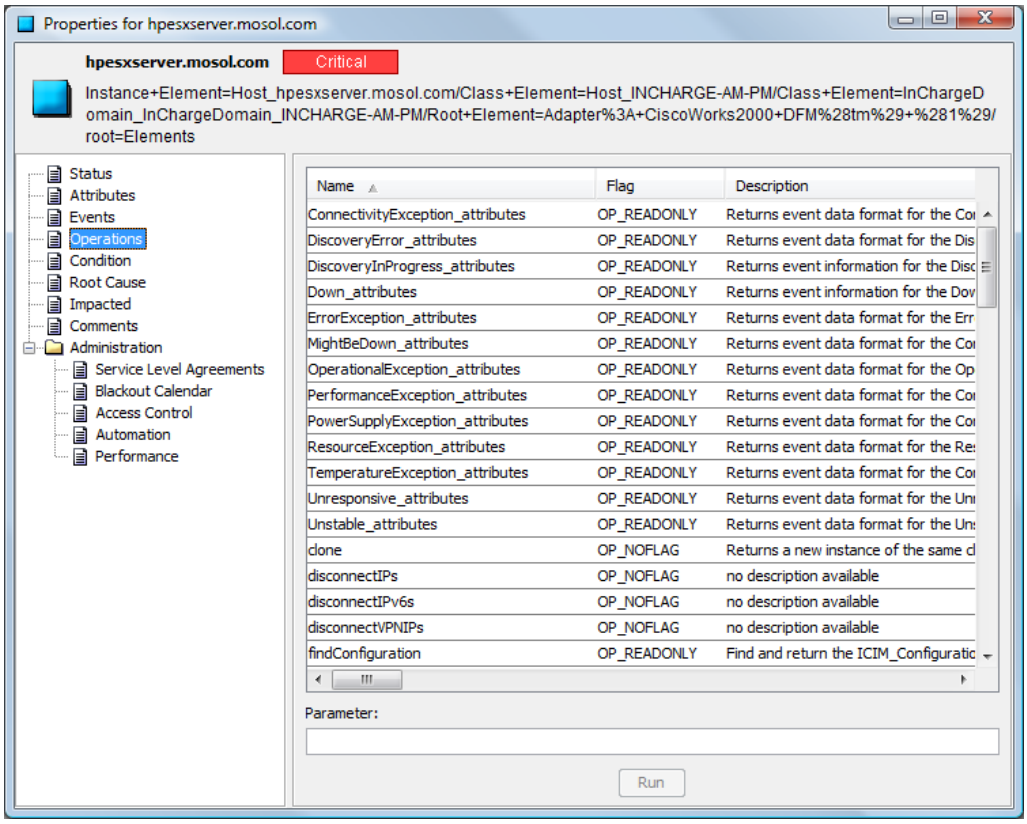
3.8.8 Running Instance Operations

Operation results (if any) appear in the Operations Center log file as a DEBUG message.

To run an operation on an instance:

1. In the *Explorer* pane, right-click an instance element under the CiscoWorks2000 DFM element, then click *Properties* to open the Status property page.
2. In the left pane, click *Operations*.

   The Operations property page opens all available operations for the instance:



   Following are descriptions of the operation flag types:

   OP_CONST: Operation does not change the object.

   OP_READONLY: Operation has no side effect on the object.

   OP_NOFLAG: Operation returns the same value if called repeatedly, with no other actions performed on the object.
3. Select an operation to highlight it.
4. Enter a parameter value in the *Parameter* field, if the operation supports it.

3.8.9 Saving and Restoring the Element Repository

Restore (read) or save the inventory of the DFM server to a file on the DFM server machine:

- "Saving the Repository" on page 56
- "Restoring the Repository" on page 56

Saving the Repository

To save the element repository:

1. In the *Explorer* pane, expand the *Administration* > *Adapters*.
2. Right-click the *CiscoWorks2000 DFM* adapter, then click *Repository Save* to open the Repository Save dialog box.
3. Specify a file name and click *OK* to save the repository information in the `SM_BASEDIR/smarts/repose` directory.

   The preferred file name extension is `.raps`.

Restoring the Repository

To restore the element repository:

1. In the *Explorer* pane, expand the *Administration* > *Adapters*.
2. Right-click the *CiscoWorks2000 DFM* adapter, then click *Repository Restore* to open the Repository Restore dialog box.
3. Specify a file name and click *OK* to restore the repository information from the specified file in the `SM_BASEDIR/smarts/repose` directory.

3.8.10 Understanding Event Notifications and Alarms Mappings

The adapter converts DFM event notifications to Operations Center alarms and displays them in the *Alarms* view.

Operations Center represents symptoms as minor alarms and compounds as major alarms (DFM does not subscribe for problems). Table 3-7 outlines these alarm mappings.

Table 3-7 *Mapping between CiscoWorks2000 DFM Notifications and Operations Center Alarm Severity*

Event	Event Type	Severity	Default Severity Color
SYMPTOM	MR_EVENT	MINOR	Yellow
COMPOUND	MR_AGGREGATION or MR_PROPAGATED_AGGREGATION	MAJOR	Orange

The adapter maps SmRemoteObserver EVENT_NOTIFY and INFORMATIONAL event types to Operations Center alarms. The configured notification mapping of the adapter instance determines the severity. The adapter always maps SmRemoteObserver EVENT_CLEAR message types to OK severity alarms in Operations Center. This cannot be changed.

Events in DFM can change the state of an alarm from active (when the event type is EVENT_NOTIFY) or inactive (when the event type is EVENT_CLEAR). Operations Center represents active alarms that became inactive with the OK alarm severity. An alarm is considered inactive when the adapter receives a SmRemoteObserver EVENT_CLEAR message type from DFM for a preexisting event.

Figure 3-4 shows Operations Center converting event notifications into alarms:

Figure 3-4 *Alarms View*

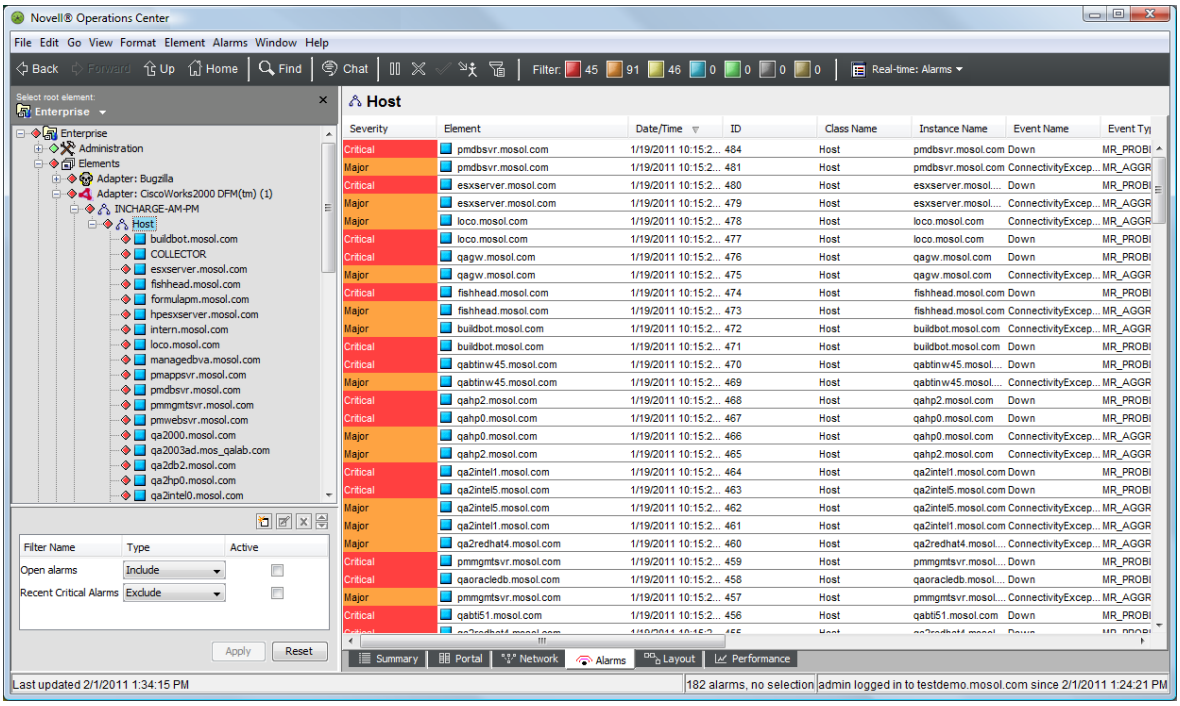


Table 3-8 lists the operations available for CiscoWorks alarms in the Operations Center *Alarms* view.

Table 3-8 *CiscoWorks2000 DFM—Alarm Operations*

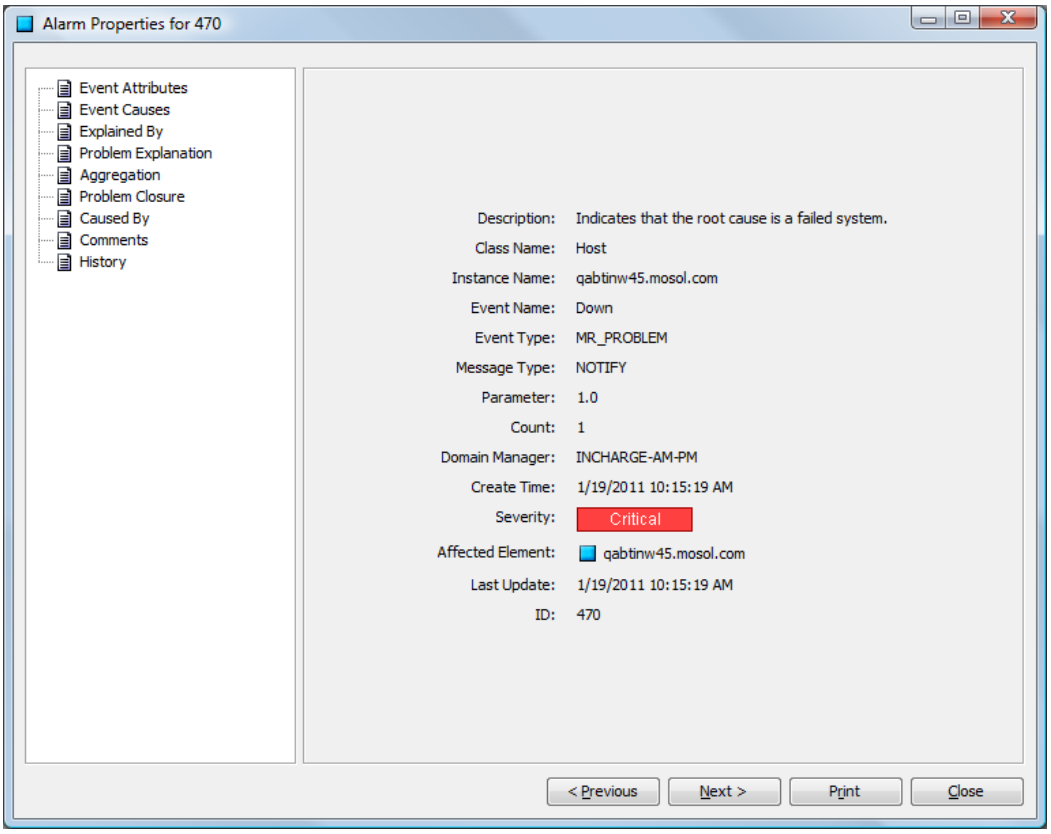
Operation	Description	Required Permission
Delete	Removes an active or inactive alarm.	Manage
Reset Counter	Resets the counter to the number of times that the Operations Center server received the event notification.	Define
Subscribe	If unsubscribed, subscribes to events with the same event name for the instance.	Define
Unsubscribe	If subscribed, unsubscribes from future events using the same event name for the instance.	Define

Operations that CiscoWorks2000 DFM does not permit do not display in Operations Center.

3.8.11 Alarm Properties

The alarm property pages for each alarm displays basic properties of the event notification, a simple description and a list of symptoms, if available.

Figure 3-5 *Alarm Events Attributes Properties for a CiscoWorks2000 DFM Alarm*



To view the properties of an alarm:

1. Right-click an alarm in the *Alarms* view, then click *Properties* to open the Status property page.
2. In the left pane, click *Event Attributes*.

   The Event Attributes property page opens and displays the properties of the alarm.
3. In the left pane, click *Description* to view a description of the alarm.
4. In the left pane, click an information page (*Event Causes*, *Explained By*, *Problem Explanation*, *Aggregation*¸ or *Problem Closure*).

   Information displays in the associated property page if the data is applicable to that alarm.

3.8.12 Subscribing to Events

The CiscoWorks2000 DFM adapter enables subscribing to and correlating events. Use the adapter to subscribe or unsubscribe to event notifications during startup or runtime.

Event subscriptions reside in a file in the `/OperationsCenter_install_path/database` directory on a per-adapter instance. If the file does not exist, a default subscriptions file is created. The file name consists of concatenating `SmartsConfig` and the adapter name.

Starting the adapter also starts the process of reading and parsing this subscription file and sending a request to DFM for each entry.

By default, event subscriptions only apply to existing instances. They do not apply to new instances that occur after a subscription profile is added, unless the *Sticky* option is selected when the subscription is created.

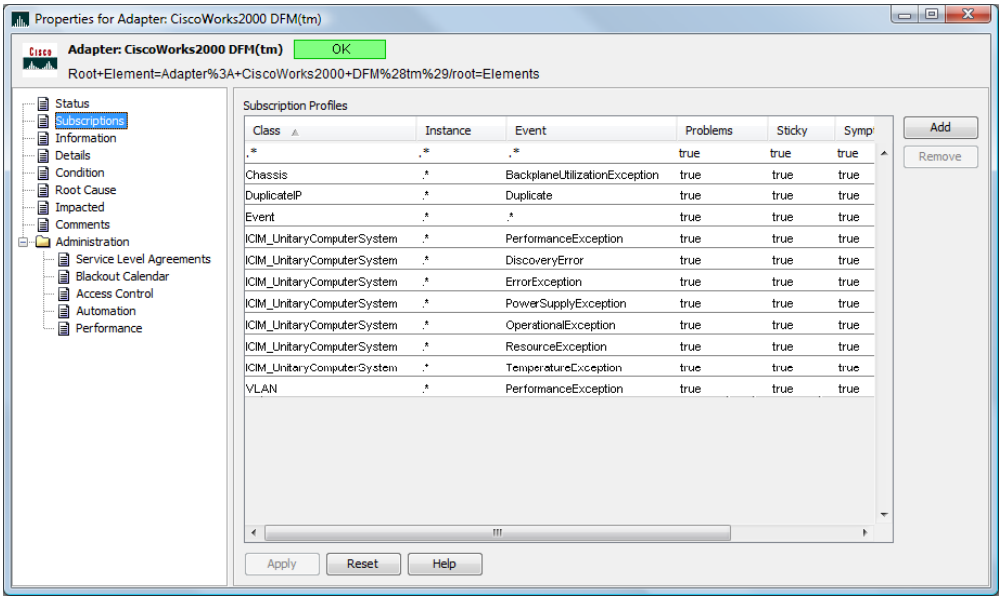
The following topics describe managing event subscriptions:

- "Setting Up Global Subscriptions" on page 59
- "Subscribing and Unsubscribing to Events at the Instance Level" on page 60
- "Subscribing and Unsubscribing to Events Directly from an Alarm" on page 61

Setting Up Global Subscriptions

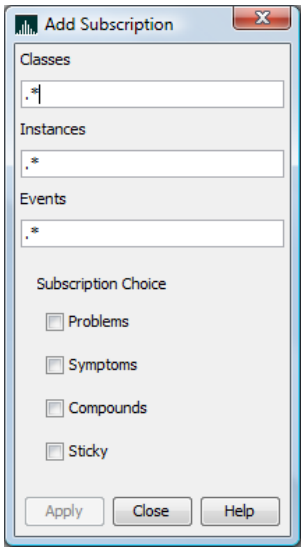
To subscribe to event notifications by setting up global subscriptions:

1. In the *Explorer* pane, expand the *Administration* > *Adapters*.
2. Right-click *CiscoWorks2000 DFM*, then do one of the following:
   - Click *Add Subscription*.
   - Click *Properties* to open the Status property page, and do the following:
     1. In the left pane, click *Subscriptions* to open the Subscriptions property page:



     2. Click *Add*.

The Add Subscription dialog box displays:



3 Specify a value or expression for the *Classes*, *Instances*, and *Events* to which a subscription is made.

   Valid syntax includes specific classes, instances, and event types. It is also possible to use regular expression syntax to match a specified pattern.

4 Select one or more subscription types:

   Problems: Notifications that pinpoint the exact cause of a failure.

   Symptoms: Notifications indicating an exceptional condition.

   Compounds: Notifications that identify one or more failures that have occurred on the same element. These notifications list the symptoms or problems that affect the element.

5 Select *Sticky* to apply this event notification subscription to all instances created (before and) after this subscription.

6 Click *Apply* to initiate the property subscription.

Subscribing and Unsubscribing to Events at the Instance Level

To subscribe and unsubscribe to event notifications at the instance level:

1 Right-click an instance in the *Explorer* pane, then click *Properties* to open the Status property page.

2 In the left pane, click *Events* to open the Events property page.

3 Do any of the following:

- To subscribe to specific events, select the events, then click *Subscribe*.
- To subscribe to all events, click *Subscribe All*.
- To unsubscribe, select the events, then click *Unsubscribe*.
- To unsubscribe to all events, click *Unsubscribe All*.

4 Close the property pages.

Subscribing and Unsubscribing to Events Directly from an Alarm

To subscribe and unsubscribe to event notifications directly from an alarm level:

1 Right-click an event in the *Alarms* view, then click *Subscribe* or *Unsubscribe*.

If the alarm represents an event that is no longer subscribed to, the Subscribe operation is enabled and the Unsubscribe operation is disabled. If the event is already subscribed to, the Subscribe operation is disabled and the Unsubscribe operation is enabled.

3.8.13 Correlating Events

The Correlation property page for the CiscoWorks2000 DFM adapter element displays various correlation metrics:

- "Viewing the Correlation Property Page" on page 61
- "Correlating Immediately" on page 61

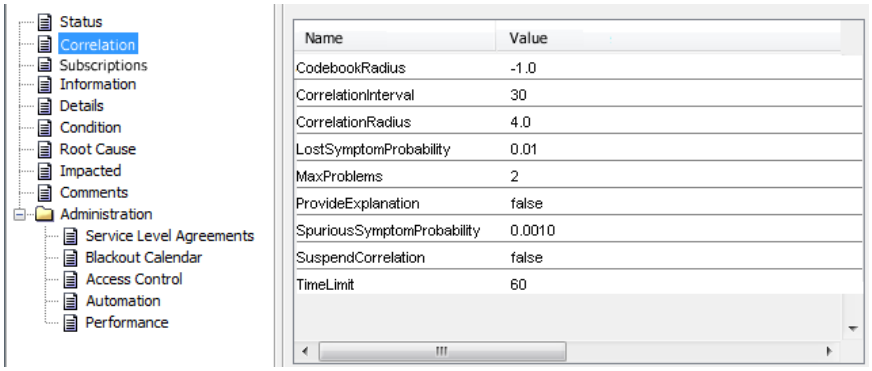
Viewing the Correlation Property Page

To view the event correlation property page

1 In the *Explorer* pane, expand *Elements*.

2 Right-click the CiscoWorks2000 DFM adapter element, then click *Properties* to open the Status property page.

3 In the left pane, click *Correlation* to open the Correlation property page:



Correlating Immediately

To activate real-time correlation for an adapter element:

1 In the *Explorer* pane, right-click the CiscoWorks2000 DFM adapter element, then click *Correlate Now*.

The Domain Manager starts the event correlation process, which continues to run in the background until it finishes.

3.8.14 Recomputing the DFM Codebook

The Codebook is a casualty mapping between problems and symptoms that is computed by the CiscoWorks2000 DFM Domain Manager's correlation engine. Use Operations Center operations to request a consistency update from the Domain Manager to regenerate the correlation codebook.

To recompute the codebook:

1 In the *Explorer* pane, right-click the CiscoWorks2000 DFM adapter element, then click *Recompute Codebook*.

The Domain Manager performs the regeneration of the correlation cookbook as a background process.

3.9 Computer Associates (CA) Spectrum

This adapter was previously named Aprisma SPECTRUM, and has the same functionality as the former one.

Create an adapter for each instance of Spectrum on the network (see Section 2.1, "Creating an Adapter," on page 17). Also modify the adapter properties (see Section A.10, "Computer Associates Spectrum," on page 295).

The Spectrum integration automatically instantiates a copy of the Visigenic ORB for the sole purpose of communicating with the SpectroServers. It is necessary to obtain a series of Spectrum CORBA API JAR files directly from Computer Associates and copy them to the Operations Center installation directory.

IMPORTANT: When creating an adapter, select the CA Spectrum adapter. Only select the older Aprisma SPECTRUM driver if upgrading from a previous release.

Refer to the following topics to integrate to Spectrum:

- Section 3.9.1, "Integrating Spectrum," on page 62
- Section 3.9.2, "Integrating Spectrum Event Descriptions," on page 65
- Section 3.9.3, "Enabling the CORBA Naming Service," on page 65
- Section 3.9.4, "Understanding Spectrum Adapter Features," on page 66

3.9.1 Integrating Spectrum

To integrate spectrum:

1 If integrating to Spectrum 9.3, verify the Operations Center server is running JRE 1.7.

2 Obtain the following files from Computer Associates, then copy them to the directories listed:

For	Obtain These Files	Copy To

Spectrum 8.0	global80.jar lm.jar omi80.jar productsuite80.jar sbgwimport80.jar snmpsrv80.jar ssorb80.jar ssorbbeans80.jar ssorbutil80.jar util80.jar utilapp80.jar utilgui80.jar utilnet80.jar utilsrv80.jar vbhelper80.jar vbjorb.jar vbsec.jar	*/OperationsCenter_install_path/* integrations/ext/spectrumV80
Spectrum 9.0	global90.jar jsafeJCEFIPS.jar lm.jar ssorb90.jar ssorbutil90.jar utilapp90.jar utilgui.jar utilsrv90.jar vbhelper90.jar vbjorb.jar	*/OperationsCenter_install_path/* integrations/ext/spectrumV90
Spectrum 9.2	global92.jar jsafeJCEFIPS.jar lm.jar ssorb92.jar ssorbutil92.jar utilapp92.jar utilgui.jar utilsrv92.jar vbhelper92.jar vbjorb.jar	*/OperationsCenter_install_path/* integrations/ext/spectrumV90
Spectrum 9.3 and 9.4	global93.jar cryptojFIPS.jar lm.jar ssorb93.jar ssorbutil93.jar utilapp93.jar utilgui93.jar utilsrv93.jar vbhelper93.jar vbjorb.jar	*/OperationsCenter_install_path/* integrations/ext/spectrumV93

3 If upgrading an existing installation, do the following:

- Delete `spectrum.jar` from the */OperationsCenter_install_path*`/integrations` directory.
- Delete all Spectrum API JAR files from the */OperationsCenter_install_path*`/classes/ext` directory.

  The table in Step 2 lists these JAR files.

4 For Spectrum version 9.0, 9.2, and 9.3, do the following:

- Copy the *SpectrumInstallation*`/tomcat/webapps/spectrum/lib/clienttopo.jar` file into the */OperationsCenter_install_path*`/integrations/ext/spectrumV90` directory.
- Copy the contents from *SpectrumInstallation*`/tomcat/webapps/spectrum/WEB-INF` to the Operations Center server. Specify this location in the *Spectrum OneClick Topology Config Directory* adapter property when you create an adapter.
- Copy the Spectrum Cause files from *SpectrumInstallation*`/SG-Support/CsPCause` to the Operations Center server. Specify this location in the *Spectrum Cause File Directory* property when you create an adapter.
- For more information about adapter properties, see Section A.10, "Computer Associates Spectrum," on page 295.

5 Set up the Spectrum SpectroServer to allow connections from the Operations Center server.

Use the Spectrum Control Panel or edit the `.hostrc` file in the Spectrum installation directory. Since Spectrum security requires this change, perform this step for each SpectroServer connected to Operations Center.

In the `.hostrc` file, add a line that contains the IP address of the Operations Center server. Do not stop the SpectroServer; changes update approximately within a minute.

The Spectrum User property value (see Section A.10, "Computer Associates Spectrum," on page 295) for the Spectrum adapter must match the IP address entered in the `.hostrc` file.

6 Enable the Corba Naming service on each Spectrum server to allow ORB independence and communication with the Operations Center server.

For more information, see the Spectrum documentation.

7 Restart the Operations Center server.

For instructions, see "Manually Starting the Operations Center Server" and "Starting the Operations Center Server in UNIX" in the *Operations Center 5.5 Server Installation Guide*.

8 Create an adapter for each instance of Spectrum on the network.

Because Spectrum requires separate sets of CORBA API libraries for Spectrum versions 8.*x*, and 9.0; separate adapter types exist to support both Spectrum versions 8.*x*, and 9.0. You must create a separate adapter for each version of the Spectrum server.

Adapters can be created for each Spectrum server, or an adapter can connect to a SpectroServer designated as a master catalog. When doing the latter, other SpectroServers must be added to the topology of the master catalog for them to show up in the adapter. For more information on adding SpectroServers to the master catalog using the Spectrum OneClick Console, consult the Spectrum documentation.

A separate Spectrum adapter license is required for each SpectroServer that Operations Center connects to, even though a single adapter can connect to multiple SpectroServers.

9 Modify adapter properties. For property descriptions, see Section A.10, "Computer Associates Spectrum," on page 295.

3.9.2 Integrating Spectrum Event Descriptions

To integrate event descriptions from Spectrum:

1. Copy the contents of the Spectrum `SG-Support` directory to a directory named `SG-Support` on the Operations Center server.
2. Set the adapter property Spectrum installation Directory (see Section A.10, "Computer Associates Spectrum," on page 295) to the parent directory of the `SG-Support` directory.

   For example, if the directory is `/opt/formula/SG-Support`, then set the property to `/opt/formula`.

   After upgrading Operations Center, it is necessary to re-create all existing Spectrum adapter instances, to allow the Spectrum adapter to dynamically load the libraries for all existing adapters.

   Subsequent sections provide more details.

3.9.3 Enabling the CORBA Naming Service

When integrating Spectrum, note that the CORBA naming service is disabled by default. To enable the service, modify the `<$SPECROOT>/lib/SDPM/partslist/NAMINGSERVICE.idb` file so that the process automatically starts the Visibroker naming service.

Enabling the CORBA Naming Service

To enable the CORBA naming service:

1. Edit the `<$SPECROOT>/lib/SDPM/partslist/NAMINGSERVICE.idb` file.

   Change the line from:

   ```
   AUTOBOOTSTART;n;
   ```

   to:

   ```
   AUTOBOOTSTART;y;
   ```

2. Add the following line to both the `<$SPECROOT>/SS/.vnmrc` and `<$SPECROOT>/SS/DDM/.configrc` files.

   For CsCLocServMapInt, add the `use_naming_service` line to `<$SPECROOT>/LS/.locrc`:

   ```
   use_naming_service=true
   ```

3. Restart the SpectroServer and ArchMgr.
4. Shut down Spectrum and reboot the system.

Verifying the Naming Service Advertisements

To verify naming service advertisements:

1. Navigate to `<$SPECROOT>/bin/VBNS` and run:

   ```
   nsutil list <SS hostname>
   ```

which returns the following:

```
Bindings in <SS hostname>
Object : CsCAlarmDomain
Object : CsCEventDomain
Object : CsCStatisticDomain
Object : CsCTypeCatalogTranslation
Object : RTMDomain
Object : CsCGlobalModelDomain
Object : CsCModelDomain
Object : CsCModelDomainTranslation
```

3.9.4 Understanding Spectrum Adapter Features

This section describes how Operations Center handles and displays the following Spectrum features:

For details on these features, see your Spectrum documentation.

Distributed Support

The Spectrum+ integration can follow remote landscape models in Spectrum. A Spectrum administrator who wants to create a link to another SpectroServer can create a remote model object. The Spectrum+ integration follows these links, contacts the remote SpectroServer, and continues mining models. Integrate any number of SpectroServers, subject to licensing.

Property Pages

Operations Center retrieves the following property pages for Spectrum elements:

- General Information
- Asset Information
- Spectrum Modeling Information
- VLAN Configuration

Model Alarms

Operations Center can retrieve and display alarms that pertain to a Spectrum model object.

Landscape Alarms

Operations Center can retrieve and display alarms set on a landscape.

Alarm Causes

The alarm property pages display probable cause text for the alarm, if the probable cause information is made available to the integration (via NFS link or copying it locally).

Topology Layouts

The *Layout* view displays topology views that are similar to SpectroGraph displays. Operations Center can display models and their positions and the links between them. However, Operations Center does not display the more complex pictographs that represent composite sites.

Logged Attributes

All attributes in the SpectroServer tagged with the logged attribute are recorded over time. However, few attributes contain this type of information. The Operations Center *Performance* view can display this data.

Model Lifecycle

The integration updates to match the model object lifecycles as Spectrum creates and destroys models.

Dynamic Discovery

Operations Center mines only the portions of the SpectroServer database that are relevant to Service Views. This promotes efficient mining by the integration.

Configuring Remote Landscape Navigation

To navigate remote landscapes:

1 In the Spectrum `agentaddr` file, specify the host names and IP addresses of remote agents.

3.10 Computer Associates Unicenter

This section refers to integrating CA Unicenter TNG or NSM products. CA Unicenter does require an ORB in order to integrate with Operations Center.

Refer to the following topics to integrate to Unicenter:

3.10.1 Integrating Unicenter

To integrate CA Unicenter and Operations Center:

1. Install the UniORB software on a supported system that has network access to the Unicenter system.

   For instructions, see Chapter 10, “ORB Installation,” on page 249.

2. Create a Unicenter adapter for each instance of UniORB on the network.

   For instructions, see Section 2.1, “Creating an Adapter,” on page 17.

3. Modify the adapter properties. For property descriptions, see Section A.11, “Computer Associates (CA) Unicenter,” on page 298).

   In the adapter’s properties, specify the Unicenter server’s hostname, UniORB port number (1580, or the one chosen earlier), and the repository name (case sensitive).

   UniORB installs as a service under Windows. Typically, the CA Unicenter Enterprise Management System, Unicenter Enterprise Manager and Unicenter WorldView are installed on the same system. If the Unicenter Enterprise Manager exists on a different machine than Unicenter WorldView, the UniORB service must run logged in as an Administrator to access enterprise event manager information.

   For a detailed description of adapter properties, see Section A.11, “Computer Associates (CA) Unicenter,” on page 298.

3.10.2 Masking Unwanted Alarms

Operations Center can compress duplicate messages into a single line. However, in some cases, it is desirable to prevent a subset of Unicenter alarm messages from ever reaching the Operations Center server.

For example, configure Unicenter to receive SNMP traps, which consist of a very large number of messages that are not associated with any object in Unicenter. As a result, these messages are also not associated with any Unicenter object displayed in the Operations Center console.

To prevent the UniORB from sending some messages to Operations Center:

1. Create a file named `uniorb.flt` in the same directory as the UniORB.

   This file contains one line representing the unchanging portion of each message that should be masked.

   Each Unicenter message that the UniORB receives is matched against each line in the `uniorb.flt` file. It drops all incoming messages that contain the complete contents of any line in the `uniorb.flt` file without sending them to the Operations Center server.

2. Verify that the lines contained in the `uniorb.cfg` file have sufficient detail so they do not unexpectedly match other alarm messages.

   This ability to eliminate messages that have no value to the user exists exclusively to ease the load on the Operations Center server and Operations Center console when maintaining and displaying messages.

3.10.3 Lazy Discovery and Show/Hide Details

For purposes of efficiency, Operations Center currently implements a Lazy Discovery for Unicenter. This means that it discovers only Unicenter World-View objects during the initial pass.

To discover all lower level (DSM) objects:

1 In the *Explorer* pane, right-click the lowest level World-View object, then click *Details* > *Show/Hide*.

The lower-level objects display.

The exceptions to this are DSM level objects that are referenced in views created using the *Service Models* element. The Unicenter adapter automatically discovers these objects during its initialization.

Operations Center initially associated alarms created by DSM level processes within Unicenter with the server object which contain them.

3.10.4 Propagate Flag Affects Parent Object Colors

Operations Center correctly supports the Propagate flag setting for World-View objects within the Unicenter 2-D map. Disabling the status propagation for any World-View object results in the parent object no longer reflecting the state of that child object. Disabling status propagation for all child objects displays the state of the parent in its natural state.

3.10.5 Child Object Propagate_Status Flag Affects Parent Color

If an object's condition color does not appear to be correct in the Operations Center console, it might be because of the presence of child objects that have other colors. Check each child object's `propagate_status` flag.

- "Checking the propagate_status Flag" on page 69
- "Enabling or Disabling the propagate_status Flag" on page 70

Checking the propagate_status Flag

To check the `propagate_status` flag of an element:

1 In the *Explorer* pane, right-click an element, then click *Properties* to open the Status property page.

If the `propagate_status` value is `Yes`, the status propagation is enabled. If it is `No`, the status propagation is disabled. The parent of the object no longer reflects the state of that child.

Enabling or Disabling the propagate_status Flag

To enable or disable the `propagate_status` flag of an element:

1 In the *Explorer* pane, right-click an element, then click one: *Propagate*, *On*, or *Off*.

3.10.6 Object Status between Unicenter and Operations Center

Status is a field in Unicenter that contains a predefined text string, showing the status of an object the last time Unicenter updated the state of the object. Operations Center supports modification of this property. Use the *Severity* right-click menu option to change the state of a Unicenter object.

For more information, see Section 3.10.9, "Comparing the Object State between Operations Center and Unicenter," on page 71.

3.10.7 Annotate Context Menu Option

The *Annotate* right-click menu option enables annotating alarm messages within Unicenter. Annotated alarms have a check mark in the *Annotate Alarm* column.

3.10.8 Operations Center Object Color versus Unicenter Object Color

Unicenter maintains two state values for each World-View object: the natural state of the object and the state of that object as determined by rolling up (propagating) the state of all child objects. Operations Center provides information about both states, but it is important to understand the differences between the two.

The color of a Unicenter object icon displayed in the Operations Center console is its propagated state. Unicenter determines the state of the object by propagating the state of all child objects that have the propagation flag enabled.

The color of an object displayed in the Operations Center console might not match the object's Severity property. Operations Center displays the propagated status using the color of the object icon that matches the color shown on the NSM 2-D map.

To check the value of a Unicenter object icon displayed in the Operations Center console:

1 In the *Explorer* pane, right-click a Unicenter element, then click *Properties* to open the Status property page.

2 Examine the *propagated_sev* field.

It contains a numeric value which Operations Center maps to a color for display purposes.

The following table shows the severity codes and their corresponding numeric and color values. The numeric value corresponds to the value in the *propagated_sev* field.

Status	Status Color	Numeric Value
OK	Green	5
INFORMATIVE	Blue	4
MINOR	Yellow	3
MAJOR	Orange	2
CRITICAL	Red	1
UNKNOWN	Gray	0

The color of a Unicenter World-View object displayed in the Unicenter 2-D Map is also based on its propagated state. Operations Center and Unicenter always agree on the color of these two objects, using the color ranges supported by Operations Center (listed in Table 3-9 on page 72).

The color severity of a Unicenter object icon displayed in the Unicenter object's Status property page is the natural state of the individual object, and might be a different color than the Unicenter object icon displayed in the Operations Center console. This can be true because Unicenter allows objects that do not monitor anything, but serve as a container for other objects. Thus, the severity of the container object might be green (Normal), while its propagated state (propagated_sev) is red (Critical).

3.10.9 Comparing the Object State between Operations Center and Unicenter

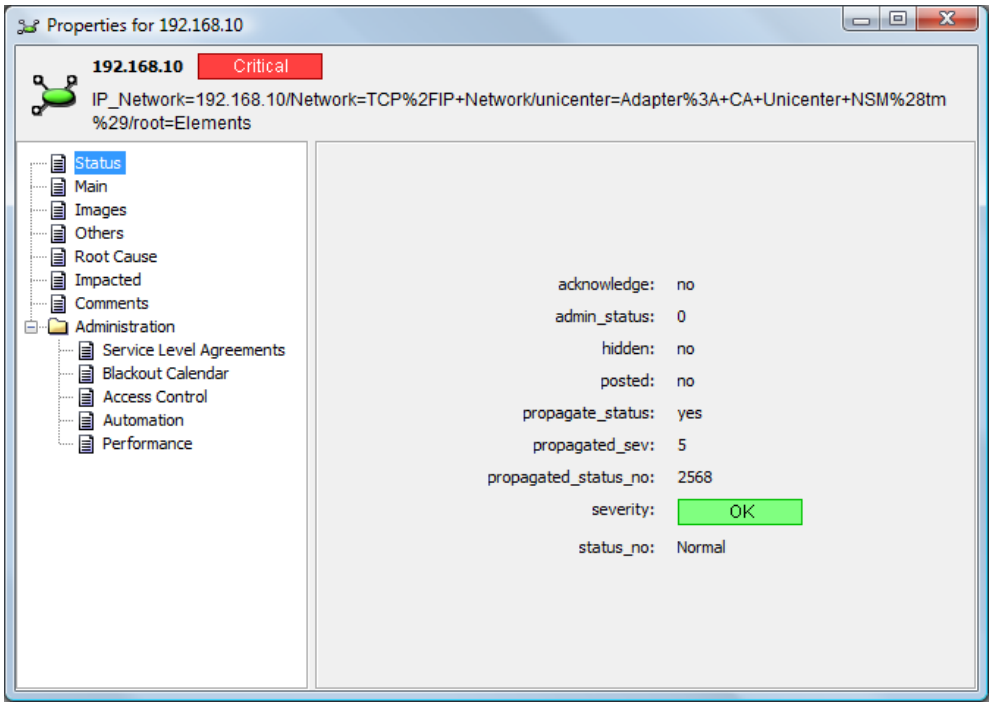
Viewing the Complete Picture of the State of a Unicenter Object

To view the complete picture of the state of a Unicenter object through Operations Center:

1. Note the color of the element displayed in the Operations Center console. In this example, assume the element is red, representing the propagated status; children with red (CRITICAL) severity roll up into this parent element.
2. Right-click the element in the *Explorer* pane, then click *Properties* to open the Status property page.

3 Note the values.

In the following figure, the green (OK) severity represents the natural status of the parent object.



Viewing the Complete Picture of the State of an Object in Unicenter

To view the complete picture of the state of an object in Unicenter:

1 Note the color of the Unicenter object.

2 Click *Open Details* on the context menu in the Unicenter 2-D map.

3 Display the *Status* tab and note the values.

Table 3-9 and Table 3-10 show the corresponding property values and colors used by Operations Center and Unicenter. Note that the Unicenter values are default values and can be customized.

Table 3-9 *Operations Center and Unicenter Corresponding Property Values*

Operations Center		**Unicenter**	
Property	**Value**	**Property**	**Value**
propagate_status	Yes or No	*Propagate*	Selected or deselected.
propagated_sev	0–9	N/A	
propagated_status_no	0–9	N/A	Default Unicenter Value.

Operations Center		Unicenter	
Property	**Value**	**Property**	**Value**
severity	critical	Severity	CRITICAL
			down
	major		MAJOR
	minor		MINOR
			WARNING
	OK		NORMAL
	informational		INSERVICE
	unmanaged		FUTURE
	unknown		UNKNOWN
			REMOVE
status_no	It might appear to be a free form text field, but is not. For more information, see Table 3-10 on page 73.	Status	String value is based on the Unicenter Alarm Set for the object.

Table 3-10 *Operations Center and Unicenter Corresponding Colors and Values*

Operations Center		Unicenter	
Color	**Value**	**Default Color**	**Default Value**
Blue	Informational	Black	Inservice
Brown	Unmanaged	Black	Future
Gray	Unknown	Black	Remove
Gray	Unknown	Gray	Unknown
Green	Normal	Green	Normal
Orange	Major	Orange	Major
Red	Critical	Red	Critical
Red	Critical	Black	Down
Yellow	Minor	Yellow	Minor
Yellow	Minor	Yellow	Warning

3.10.10 Support for DSM Object Level Alarms

In Operations Center, the alarms for DSM level objects in the Unicenter repository display in the context of the actual DSM-level process that created them, and are color-coded according to their severity.

Operations Center can display objects in the Unicenter Common Object Repository (COR) that are not visible in the Unicenter 2-D map. These DSM-level objects include specific instances of the processes, logs, file systems, and so on, that Unicenter agents monitor and display as child objects of the Unicenter agents that define and control them.

3.11 EMC SMARTS

Create an adapter for each instance of EMC SMARTS on the network (see Section 2.1, "Creating an Adapter," on page 17). See the integration information in the following sections and then modify the adapter properties (see Section A.12, "EMC SMARTS," on page 300).

The following topics provide information on integrating to EMC SMARTS:

3.11.1 Integrating EMC SMARTS

To integrate to EMC SMARTS:

1. Stop the Operations Center server.

   For instructions, see "Stopping the Operations Center server in Windows" and "Stopping the Server and the mosdaemon manually in UNIX" in the *Operations Center 5.5 Server Installation Guide*.

2. Verify that the Operations Center server is not configured to restart automatically.

   For more information, see the *Operations Center 5.5 Server Configuration Guide*.

3. Copy the following files from the EMC SMARTS installation `/classes` directory into the `/`*`OperationsCenter_install_path`*`/classes/ext` directory:

   - `clsapi.jar`
   - `clsapi_err.jar`

- `cryptojFIPS.jar`
- `i18napi.jar`
- `icu4j-3_8_1.jar`
- `net.jar`
- `platform.jar`
- `platform_err.jar`
- `skclient.jar`
- `skclient_err.jar`
- `sslj.jar`

4 Update or install the license file through the Operations Center Configuration Manager, if necessary.

   The license file must contain one or more key entries for `com.mosol.integration.smarts.InChargeIntegration`.

5 Restart the Operations Center server.

   For instructions, see "Manually Starting the Operations Center Server" and "Starting the Operations Center Server in UNIX" in the *Operations Center 5.5 Server Installation Guide*.

6 Create an adapter for each instance of EMC SMARTS on the network.

   For details, see Section 2.1, "Creating an Adapter," on page 17.

7 Modify the adapter properties. For property descriptions, see Section A.12, "EMC SMARTS," on page 300.

   It is helpful to know the following information:

   - The hostname of the SMARTS Broker
   - The port that the SMARTS Broker listens on (default is 9002)
   - A valid user name and password, if SMARTS is running with authentication required
   - The structure of your SMARTS environment. The discovery of instances is dependent on a match of both Domain List and Domain Manager properties. If a domain isn't subscribed to by the specified Domain Manager, Operations Center won't be able to access it.

8 Modify property files to restrict adapter functions.

   For more information, see Section 3.11.2, "Disabling Operations and Filtering Information from EMC SMARTS," on page 75.

3.11.2 Disabling Operations and Filtering Information from EMC SMARTS

The EMC SMARTS adapter can be restricted from performing various operations on the EMC SMARTS server. This includes various intrusive functions such as saving or restoring the Repository file or performing codebook correlation updates.

Operations Center uses properties files to restrict server-side access for all EMC SMARTS adapters and apply filter relation and alarm information. The following default files located in the `/OperationsCenter_install_path/database` directory apply to all adapter instances:

- **SmartsSecurityFilter.properties:** Disable various adapter operations and functions to the EMC SMARTS server.
- **SmartsAlarmFilter.properties:** Filter alarms received from EMC SMARTS.
- **SmartsRelationFilter.properties:** Filter the relations shown.

If no property file exists, no restrictions apply to the adapter.

Table 3-11 describes the default property files.

Table 3-11 *EMC SMARTS –Properties Files and Filter/Disable Options*

File Name	Entry	Description
`SmartsSecurityFilter.properties`	Codebook	Disables Consistency Update and Correlate Now operations.
	Repository	Disables Store Repository and Restore Repository operations.
	Manage	Disables Manage and Unmanage element operations.
	Instance	Disables Instance Create and Delete operations on elements.
	Topology	Disables the subscription to topology change notifications.
`SmartsAlarmFilter.properties`	`AlarmColumn=value`	Allows any alarms to pass with the specified alarm column value. Separate values with a "|" symbol and show spaces in values with an underscore (_) symbol.
`SmartsRelationFilter.properties`	`pass=relationType`	Indicates the relations to show. The default value is `pass=ServiceOffering`.
	`fail=relationType`	Indicates the relations to show, but does not show any children. Default values include: `Bridges`, `BridgedVia`, `ConnectedPorts`, `ConnectedTo`, `ConnectedVia`, `ConsistsOf`, `LayeredOver`, `MemberSystems`, `Members`, `Underlying`, `Manages`, `Consumes`, `Produces`, `Serves`, `HostedBy`, `ManagedBy`, `ConsumedBy`, `ProducedBy`, `ServedBy`, and `MemberOf`.
	`exclude=relationType`	Indicates those relations to filter out (not shown). Default values include: `ConfiguredBy`, `SettingsAppliedBy`, `Realizes`, `RealizedBy`, `Notifications`, `Notifies`, `NotifiedBy`, `NeighboringSystems`, `part of`, `MemberOf`, `ParentGroup`, `Factory`, and `ConsistsOf`.

These filter properties files can be customized for a specific adapter.

To apply a property file for a single adapter:

1. Make a copy of the property file in */OperationsCenter_install_path/*`database` and rename using the adapter name such as `SmartsfilterTypeFilter.adapterName.properties`.

   Use underscores in place of spaces within an adapter name. For example, if the adapter name is `InCharge on serverx`, rename the filter file to `SmartsAlarmFilter.InCharge_on_serverx.properties`.

2. Modify the properties file as required.

3.11.3 Error Handling

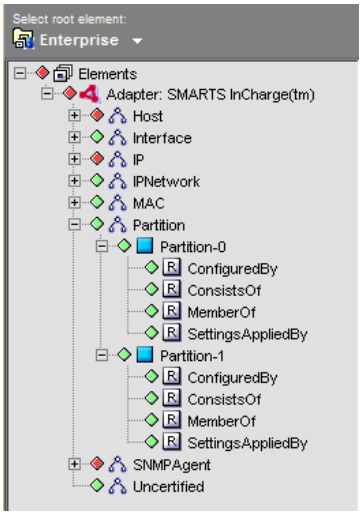
The integration might throw exceptions in the case of errors or malfunctions. All exceptions are logged in a standard format using the Operations Center logging features. Each logging statement is associated with an ERROR, INFO, WARN or DEBUG level.

3.11.4 Discovery of EMC SMARTS Elements

The EMC SMARTS adapter automatically performs element discovery synchronization that includes:

- Discovery of all the technology elements from the InCharge database and populates the Operations Center object model
- Discovery of classes, instances, and relations between objects for use in the Service Views
- Relationship mapping between object for *n*-deep nested class/instance/relation sets
- Subscriptions to event notifications
- Management of instances (when available)
- Element property mapping and property pages for elements
- Automatic adapter updates for InCharge instance create and delete notifications

Figure 3-6 *Explorer Pane: InCharge objects are listed under the adapter element*



The specific objects are implemented as proxy objects of the actual InCharge objects. When values (or other data) are queried in Operations Center, the adapter requests the data on-demand from the InCharge server. The adapter caches some common data values for efficiency.

Table 3-12 describes the InCharge object icons.

Table 3-12 InCharge Objects

Icon	Description
	The Root Element represents an instance of a Domain Manager
	Class Element
	Instance Element
	Relation Element

Elements in Operations Center are identified by a unique key value. The key value consists of a concatenation of up to 3 string values delimited by ":". This key value is used to uniquely identify the element instance in the Operations Center element repository.

Operations Center aggregates all data from various consoles into the element hierarchy. When comparing the Operations Center element hierarchy tree to the InCharge console, typically more relation instances are displayed in Operations Center than in a single InCharge console. This is because the InCharge console only shows the relations applicable to the console.

Parent a single element to multiple spots in the element hierarchy by using a key value that consists of concatenating the root hierarchy key and all element keys for each of its parents.

- Class instance key values are simply the class name. For example, Host.
- Instance key values are the concatenation of the class with the instance name. For example, Host:server.mosol.com.
- Relation instance key values are the concatenation of the class, instance, and relation name. For example, Host:server.mosol.com:ComposedOf.

Normally, the default condition algorithm determines the condition of a Operations Center element. The default condition algorithm sets the element condition equal to the most critical severity of any alarm attached to the element or to its children. The exception is for relation elements, which do not propagate up their condition.

For information on changing the default algorithm, see the *Using Algorithms to Calculate Element State* in the *Operations Center 5.5 Server Configuration Guide*.

Table 3-13 outlines the ACL permission on various element menu operations.

Table 3-13 EMC SMARTS—ACL Permissions on Element Operations

Element	Operation	ACL Permission
Adapter (top-level)	Add Subscription	Define
	Correlate Now	Manage
	Instance Create	Define
	Instance Delete	Define
	Recompute Codebook	Manage
	Repository Save	Define
	Repository Restore	Define

Element	Operation	ACL Permission
Instance	Manage	Manage
	Unmanage	Manage
	Delete	Define

3.11.5 Viewing Attributes, Details, Programs, and Libraries

The EMC SMARTS adapter property pages provides a way to review information available regarding attributes, programs, libraries and other features. The property pages provide this information in various specialized pages for the adapter, instance, and class elements.

To open the EMC SMARTS adapter property pages:

1. In the *Explorer* pane, expand *Elements*.
2. Right-click *Adapter: EMC SMARTSInCharge*, then click *Properties* to open the Status property page.

Repeat the previous steps to open the property pages for instance and class elements listed beneath the adapter element.

Table 3-14 outlines the various information and features that are found in the property pages for the adapter root, class and instance elements.

Table 3-14 *EMC SMARTS—Element Properties Information*

Element Type	Property Page	Description
Adapter Root	Details	Provides various information about the adapter connection with the SMARTS Domain Manager including status, socket port and time out settings.
	Correlation	Provides information regarding the correlation codebook and its associated settings. For more information, see Section 3.11.11, “Correlating Events,” on page 88.
	Programs	Displays the list of programs loaded in the server.
	Libraries	Displays the list of models loaded in the server.
	Subscriptions	Displays a list of event notification subscriptions (see Section 3.11.10, “Subscribing to Events,” on page 85). Allows adding, changing, or deleting subscriptions.
Class	Description	Displays a textual description for the class element.
	Attributes	Provides the values of various attributes associated with the element.
	Events	Displays a list of simple events defined for this class. The simple event names are shown. Information in this property page is read-only.
	Operations	Displays a list of all operations defined for this class.

Element Type	Property Page	Description
Instance	Attributes	Provides the values of various attributes defined for the class. *Refresh* updates the attribute values in real time. Attributes Access types: **MR_NO_ACCESS:** Attribute value is unavailable. **MR_STORED:** Attribute's value is set when the class is modeled, or instantiated. If the attribute is not defined as read-only, it can also be modified at runtime. **MR_COMPUTED:** Attribute's value is computed when the value is requested. Computed access types are always read-only. **MR_INSTRUMENTED:** Attribute's value is retrieved from an external source through a protocol. **MR_PROPAGATED:** Attribute's value is derived from the values of other class instances with which this class is involved in a relationship or relationship set. **MR_UNCOMPUTABLE:** Attribute's value cannot be computed. **MR_COMPUTED_WITH_EXPRESSION:** Attribute's value is computed when the value is requested, using an expression. Computed access types are always read-only.
	Events	Displays a list of simple events defined for this class. Allows subscribing and unsubscribing to event notifications for the instance element (see Section 3.11.8, "Event Notifications and Alarms Mappings," on page 84).
	Operations	Displays a list of all available operations for this element. Allows running a selected operation (see "Running Instance Operations" on page 82).

3.11.6 Creating and Managing Instances

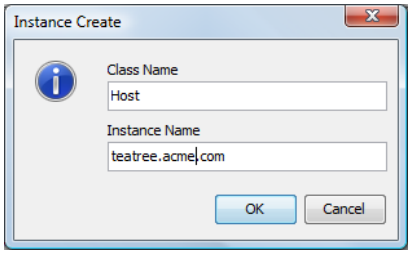
Managed instances are monitored by InCharge. Unmanaged elements are probed and associated information is stored in the InCharge inventory, but InCharge does not monitor the element.

Adding elements to or deleting elements from the Domain Manager from another client sends to the adapter notification about the event. The adapter automatically creates (or delete) the appropriate class and instance elements, allowing for a lazy discovery of object relations.

Creating an Instance

To create an instance:

1. In the *Explorer* pane, expand *Elements*.
2. Right-click the EMC SMARTS adapter element, then click *Instance Create* to open the Instance Create dialog box:



3. Specify the following:

   Class Name: Name of the class.

   Instance Name: Name of the instance.

4. Click *OK* to create the instance.

Deleting an Instance

- "Deleting by Right-Clicking" on page 81
- "Deleting Using a Dialog Box" on page 81

Deleting by Right-Clicking

To delete an instance by right-clicking:

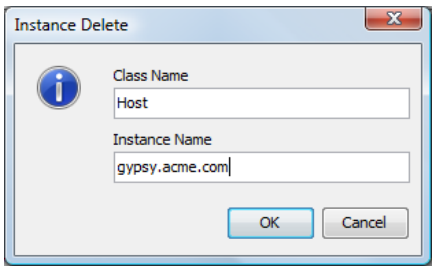
1. Right-click the instance element in the *Explorer* pane, then select *Delete*.
2. Click *Yes* when prompted for confirmation.

   The instance and its children are deleted.

Deleting Using a Dialog Box

To delete an instance using a dialog box:

1. In the *Explorer* pane, expand *Elements*.
2. Right-click *Adapter: EMC SMARTS*, then click *Instance Delete* to open the Instance Delete dialog box:



3 Fill in the fields:

Class Name: Name of the class.

Instance Name: Name of the instance.

4 Click *OK* to delete the instance.

Managing an Instance

To start managing an instance:

1 In the *Explorer* pane, expand *Elements* > *Adapter: EMC SMARTS* and a class element.

2 Right-click a manageable instance element, then click *Manage*.

Unmanaging an Instance

To unmanage an instance:

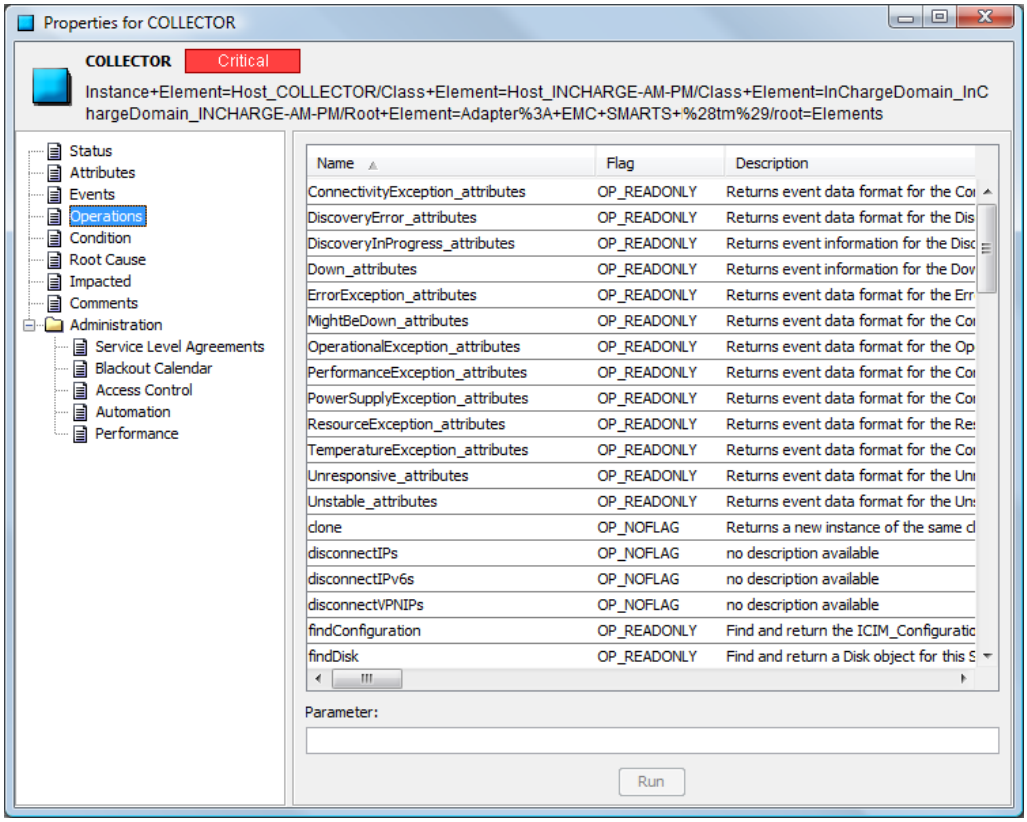
1 In the *Explorer* pane, right-click the instance element, then click *Unmanage*.

Running Instance Operations

To run an operation on a particular instance:

1 Right-click an instance element in the *Explorer* pane, then click *Properties* to open the Status property page.

2 In the left pane, click *Operations* to display all operations available for the instance.



The following are descriptions of the operation flag types:

OP_CONST: Operation does not change the object's state.

OP_READONLY: Operation has no side effect on the object.

OP_IDEMPOTENT: Operation returns the same value if called repeatedly, with no other actions on the object.

OP_NOFLAG: No other flags apply to the operation.

3 Select an operation.

The selected operation is highlighted.

4 Enter a parameter value in the Parameter text file if the operation supports it.

5 Click *Run*.

The result (if any) of the operation is written to the Operations Center log file as a DEBUG message.

3.11.7 Saving and Restoring the Element Repository

Restore the inventory of the InCharge server or save it to a file on the InCharge server machine:

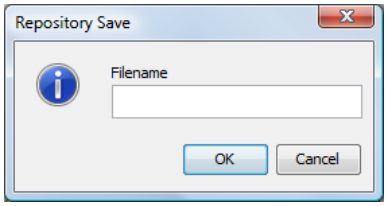
- "Saving the repository" on page 83
- "Restoring the repository" on page 84

Saving the repository

To save the repository:

1 In the *Explorer* pane, expand *Elements*.

2 Right-click *Adapter: EMC SMARTS*, then click *Repository Save* to open the Repository Save dialog box:



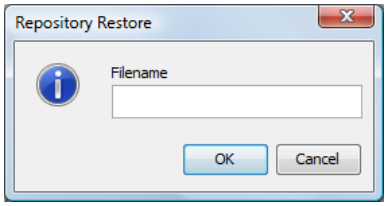
3 Specify a file name and then click *OK*.

Repository information saves to the `SM_BASEDIR/smarts/repos` directory. The preferred file name extension is `.rps`.

Restoring the repository

To restore the repository:

1. In the *Explorer* pane, expand *Elements*.
2. Right-click *Adapter: EMC SMARTS*, then click *Repository Restore* to open the Repository Restore dialog box:



3. Specify a file name and click *OK* to restore repository information from the `SM_BASEDIR/smarts/repos` directory from the specified file.

3.11.8 Event Notifications and Alarms Mappings

The adapter converts InCharge event notifications to Operations Center alarms and displays them in the *Alarms* view.

Operations Center represents symptoms as MINOR alarms, compounds as MAJOR alarms and problems as CRITICAL alarms. Table 3-15 outlines these alarm mappings.

Table 3-15 *Mapping InCharge Notifications to Operations Center Alarm Severities*

Event	Event Type	Severity	Default Severity Color
SYMPTOM	MR_EVENT	MINOR	Yellow
COMPOUND	MR_AGGREGATION or MR_PROPAGATED_AGGREGATION	MAJOR	Orange
PROBLEM	MR_PROBLEM	CRITICAL	Red

The adapter maps SmRemoteObserver EVENT_NOTIFY and INFORMATIONAL event types to Operations Center alarms. The configured Notification Mapping of the adapter instance determines the severity. The adapter always maps SmRemoteObserver EVENT_CLEAR message types to OK severity alarms in Operations Center. This cannot be changed.

Events in InCharge can change the state of an alarm from active (when the event type is EVENT_NOTIFY) or inactive (when the event type is EVENT_CLEAR). Operations Center represents active alarms that became inactive with the OK alarm severity. An alarm is inactive when InCharge sends a SmRemoteObserver EVENT_CLEAR message type to the adapter for a preexisting event.

Table 3-16 lists the available operations for alarms.

Table 3-16 *EMC SMARTS—ACL Permissions on Alarm Operations*

Operation	Description	ACL Permission
Acknowledge	Acknowledges an active alarm.	Manage

Operation	Description	ACL Permission
Unacknowledge	Unacknowledges an active alarm.	Manage
Delete	Removes an inactive alarm.	Manage
Subscribe	Subscribes to events with the same Event Name for the instance.	Define
Unsubscribe	Unsubscribes from future events with the same Event Name for the instance.	Define

Operations that are not permitted by EMC SMARTS do not display in Operations Center.

3.11.9 Alarm Properties

The alarm property pages for each alarm display basic information about the event notification, a simple description and a list of symptoms, if available.

To view the properties of an alarm:

1. Right-click an alarm in the *Alarms* view, then click *Properties* to open the Status property page.
2. In the left pane, click *Event Attributes*.

   The Event Attributes page displays the alarm properties.
3. Click *Description*.

   The Description page displays a description of the alarm.
4. Click to open other pages: *Event Causes*, *Explained By*, *Problem Explanation*, *Aggregation* and *Problem Closure*.

   Information displays in the associated property page if the data is applicable to that alarm.

3.11.10 Subscribing to Events

Use the EMC SMARTS adapter to correlate and subscribe to events. The adapter provides the ability to subscribe or unsubscribe to event notifications during startup or during run time.

Event subscriptions reside in a file in the `/OperationsCenter_install_path/database` directory on a per-adapter instance. The file name consists of concatenating `SmartsConfig` and the adapter name. If the file does not exist, a default subscriptions file is created.

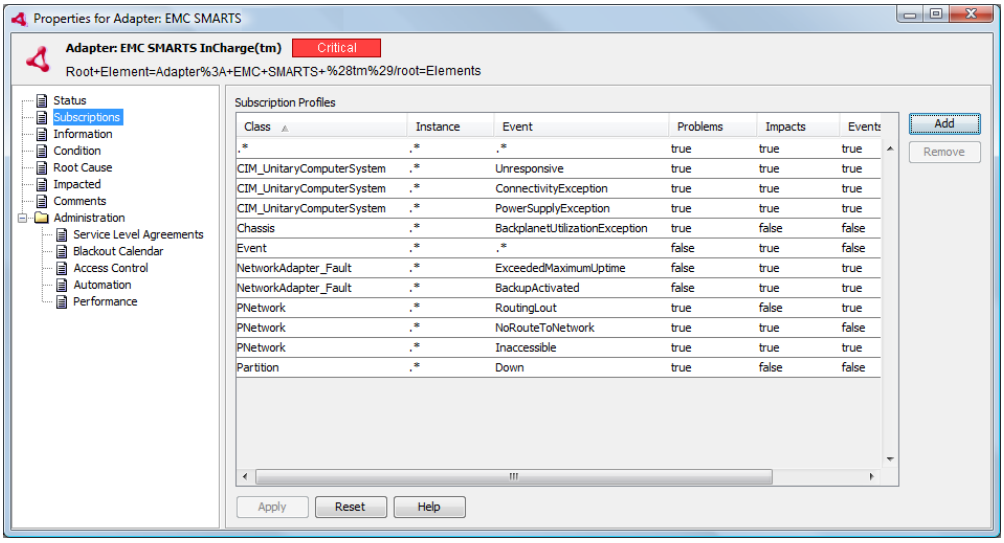
During adapter startup, reading and parsing this subscription file occurs, and a request is sent to InCharge for each entry.

- "Setting Up Subscriptions Globally" on page 86
- "Subscribing or Unsubscribing to Events at the Instance Level" on page 87
- "Subscribing to Events Directly from an Alarm" on page 87

Setting Up Subscriptions Globally

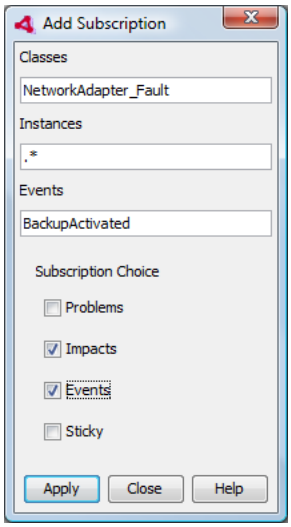
To set up subscriptions globally:

1. In the *Explorer* pane, expand *Elements*.
2. Do one of the following:
   - Right-click the *Adapter:EMC SMARTS* adapter element, then click *Add Subscription*.
   - Do the following:
     1. Right-click the *Adapter:EMC SMARTS* adapter element, then click *Properties* to open the Status property page.
     2. In the left pane, click *Subscriptions* to open the Subscriptions property page:



3. Click *Add*.

The Add Subscription dialog box is displayed:



3 Specify a value or an expression for Classes, Instances, and Events. Valid syntax includes specific classes, instance, and events or regular expression syntax that matches a specified pattern.

   The example shown in the Add Subscription dialog box (see Step 2) subscribes to all classes of events. It is also possible to subscribe to a specific class of events such as Switch. Refer to the EMC SMARTS filter properties files for additional information.

   By default, event subscriptions only apply to existing instances. They do not apply to new instances that are added later, unless the *Sticky* option is selected when the subscription is created.

4 Select one or more subscription options:

   Problems: Notifications that pinpoint the exact cause of a failure.

   Impacts: Notifications indicating an exceptional condition.

   Events: Notifications that identify one or more failures that occurred on the same element. These notifications list the symptoms or problems that affect the element.

   Sticky: Applies this event notification subscription to all instances created before and after this subscription.

5 Click *Apply* to initiate the property subscription.

Subscribing or Unsubscribing to Events at the Instance Level

To subscribe or unsubscribe to events at the instance level:

1 In the *Explorer* pane, expand *Elements* > *Adapter:EMC SMARTS* and a class element.

2 Right-click an instance element, then click *Properties* to open the Status property page.

3 In the left pane, click *Events* to open the Events property page.

4 Do one of the following:

- To subscribe to selected events, click the events and then click *Subscribe*.
- To subscribe to all events, click *Subscribe All*.
- To unsubscribe to selected events, select the events and then click *Unsubscribe*.
- To unsubscribe to all events, click *Unsubscribe All*.

Subscribing to Events Directly from an Alarm

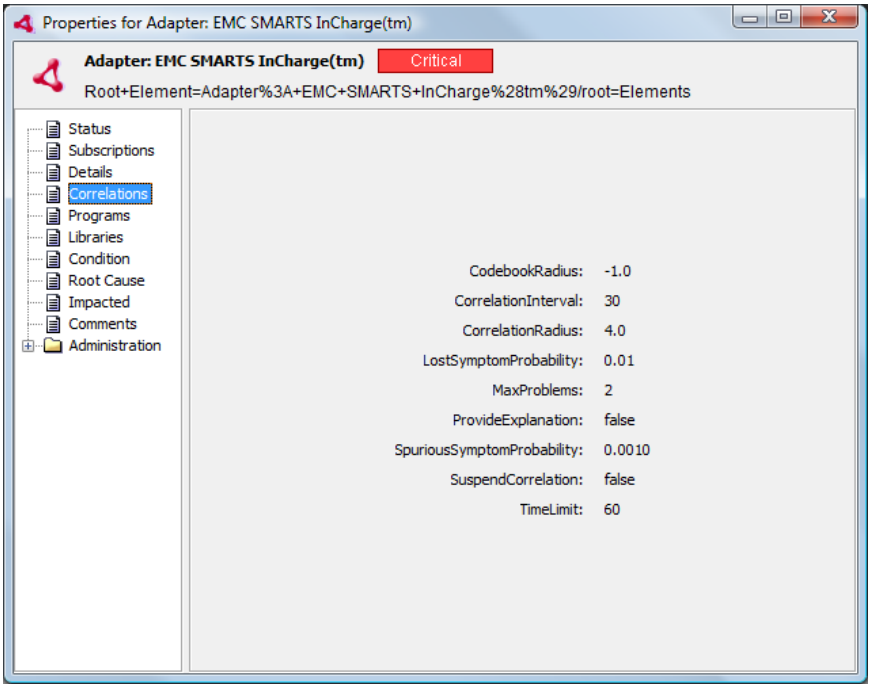
To subscribe to events directly from an alarm, right-click an event in the *Alarms* view and then click *Subscribe* or *Unsubscribe*.

The Subscribe and Unsubscribe operations are enabled or disabled on the menu depending on whether the event represented by the alarm is subscribed to. For example, if the event is subscribed to, the Subscribe operation is disabled and the Unsubscribe operation is enabled.

3.11.11 Correlating Events

The Correlation property page for the EMC SMARTS adapter element displays various correlation metrics.

Figure 3-7 *Adapter Property Pages: The Correlation page displays event correlation metrics for a EMC SMARTS element*



To correlate immediately:

1. In the *Explorer* pane, expand *Elements*.
2. Right-click the *EMC SMARTS* adapter element, then click *Correlate Now*.

   The Domain Manager starts the event correlation process which runs in the background until it finishes.

3.11.12 Recomputing the InCharge Codebook

The Codebook is a casualty mapping between problems and symptoms that computed by the EMC SMARTS Domain Manager's correlation engine. Use Operations Center operations to request a consistency update from the Domain Manager to regenerate the correlation codebook.

To recompute the codebook:

1. In the *Explorer* pane, expand *Elements*.
2. Right-click the *EMC SMARTS* adapter element, then click *Recompute Codebook*.

   The Domain Manager performs the regeneration of the correlation cookbook as a background process.

3.12 HP OpenView Network Node Manager

Operations Center offers two adapters that integrate with HP OpenView Network Node Manager software. The basic integration required an ORB installation to integrate with the software. The other, the NNMi adapter does not.

3.12.1 HP OpenView NNM

The HP OpenView (NNM) adapter works in conjunction with the OvORB. OvORB provides the following features:

- Starting and stopping automatically with NNM as an OV daemon process, not on a per-console basis. The OvORB is an OV managed process.
- Managing alarm data. The adapter can run without the OvORB to manage object, symbol and topology data.
- Lazy and Auto Discovery.
- Access to SNMP Varbinds in alarms.
- Importing NNM event definitions from the `trapd.conf` file and displaying them in Operations Center.
- Alarm Normalization Customization. Derives Object ID, Event Source, Event Time, and Severity of an event on a per Event OID basis.
- Access to all object, symbol, and submap information of elements.
- Condition algorithms on elements.
- `Manage` and `Unmanage` element commands.
- Map properties available from the root element property pages (Details page).
- Event descriptions available from the root element property pages (Event Descriptions page) and from alarm properties.
- `Acknowledge`, `Unacknowledge` and `Delete` alarm commands.

Ping and Traceroute originate from the OvORB machine. If the OvORB is not installed, they originate from the Operations Center server.

Operations Center displays graphical representations of OV symbol, submap, and connection icons. Elements in the *Explorer* pane use the OV icon to identify the class of an object, or use a connection symbol to show a connection object.

The OvORB manages all alarm and event logic. The NNM adapter is responsible for representing the map, symbol and topology data. The NNM adapter is also responsible for normalizing NNM event data into Operations Center alarms. It performs this normalization based on metadata derived from the NNM event definition and the alarm normalization data.

If the OvORB goes down for maintenance or a crash, the integration continues to run. When the OvORB restarts, the integration automatically recovers without requiring stopping and restarting the adapter. After the OvORB restarts, it automatically resynchronizes changed alarm data.

Integration without the OvORB

It is possible to integrate the HP OpenView (NNM) adapter and Operations Center without the OvORB. When configuring the OpenView adapter properties (see Section A.13, "HP OpenView Network Node Manager," on page 301), leave the adapter.orb port blank (the default). Start the NNM adapter and it should connect to the ovw map; the adapter icon should change to green. If the connection is unsuccessful, the icon is red (CRITICAL).

Setting up the Integrated Environment

Because of HP architecture limitations, the following requirements exist for proper function of the NNM integration:

- The default map must be read/write. The NNM integration always uses the default map and it requires read/write permissions. The adapter can only attach to the default map. This is a limitation of the HP NNM API.
- An NNM console must remain open for the integration to maintain a session with NNM. However, it is not necessary for this console to be the default read/write map.

The integration with HP NNM is a two-part process. It requires a one-time setup of the Operations Center server, followed by installation of the OvORB software. Even if there is no plan to manage alarms or use the OvORB services, it is necessary to install the software on the NNM machine because the adapter requires some files.

To integrate OpenView:

1. Stop the Operations Center server.

   For instructions, see "Stopping the Operations Center server in Windows" and "Stopping the Server and the mosdaemon manually in UNIX" in the *Operations Center 5.5 Server Installation Guide*.

2. Configure the Operations Center server to not restart automatically.

   For more information, see the *Operations Center 5.5 Server Configuration Guide*.

3. On the Operations Center server machine, copy (or symbolically link) the `libovw.jar` file from the HP Network Node Manager distribution to the `/OperationsCenter_install_path/classes/ext` directory.

   The `libovw.jar` file is on the NNM server machine in the `/opt/OV/www/htdocs/classes` directory.

   For OpenView 7.5, copy an additional file, `launcher/ovlaunch.jar`, to the `/OperationsCenter_install_path/classes/ext` directory. This path is relative to the location of `libovw.jar` in the HP Network Node Manager distribution.

4 Update or install the license file through the Operations Center Customizer, if applicable.

   The license file must contain one or more key entries for `com.mosol.integration.nnm.NNMntegration`.

5 Install, configure and start the OvORB on the NNM server machine.

   For instructions, see Chapter 10, "ORB Installation," on page 249.

6 Start the Operations Center server and launch the Operations Center console.

7 Create an OpenView Network Node Manager adapter for each instance of NNM.

   For instructions, see Section 2.1, "Creating an Adapter," on page 17.

8 Modify the HP OpenView Network Node adapter properties.

   For property descriptions, see Section A.13, "HP OpenView Network Node Manager," on page 301.

   The two `nnm.jar` files, the Operations Center server `integrations/nnm.jar` and the OvORB `classes/nnm.jar` should be the same. Otherwise, a warning message is issued.

Port Communications Setup

To establish communication between the NNM adapter and OvORB, use the following port assignments:

NNM Adapter Port Number	OvORB Port Number
1572	Any port
1573	Any port
Any port	1572
Any port	1573
Any port	3700–3720
Any port	2447

Upgrading a Pre-3.5 Adapter

The Operations Center upgrade process automatically upgrades all legacy adapter definitions. However, the installer only performs the upgrade on the existing `adapters.ini` file in the `/`*`OperationsCenter_install_path`*`/database` directory.

After performing a clean installation and copying the configuration files (such as `adapters.ini`) to the new installation directory, it is necessary to manually upgrade all pre-3.5 NNM adapter instances. These steps require deleting the adapter instance and creating a new NNM adapter.

Use the following steps to manually update the pre-3.5 version `adapters.ini` file. It is necessary to follow these steps only if backwards compatibility is required with a pre-3.5 version OpenView adapter instance.

To manually upgrade a pre-3.5 adapter:

1 Open the adapter properties for the pre-3.5 adapter and determine the adapter instance from the DName.

   The adapter instance is found after the `openView:` part of the DName. For example, if the DName is `openView:4=MyOV/root=Elements`, the instance is 4.

2 Delete the adapter instance.

3 Create a new NNM adapter.

   Use the same adapter name as the legacy adapter.

4 Specify the legacy adapter's adapter instance in the Adapter Instance property.

   Note that some adapter properties are no longer available in the adapter properties setting for the NNM adapter. Command.Ping and Command.Trace.Route now run in the ORB and are configured in the new OvORB. RequestDepth is not available because topology information is requested directly through the NNM API in the adapter, rather than through the ORB.

Managing and Unmanaging Elements

Manage or unmanage objects using the element menu options in the Operations Center console. Integrations using OpenView NNM version 6.0 support the `Manage` and `Unmanage` element commands.

For details on using these options or commands, see *Managing and Unmanaging Elements* in the *Operations Center 5.5 User Guide*.

Symbol Status Change Rules: Symbols representing unmanaged objects do not receive status updates from applications. All symbols representing an object on a particular map, regardless of the symbol status source, change to the UNMANAGED state if the underlying object is unmanaged.

Viewing and Updating Alarms

The Operations Center *Alarms* view displays the alarms that display in the NNM Alarm Browser. When the integration starts with a connection to the OvORB, the integration is ready to begin receiving real-time alarm data from NNM and displaying them in the *Alarms* view.

The element to which the alarm attaches is the alarm's affected element. By default, this is the object that corresponds to the alarm's NNM object ID. In cases where the NNM object ID is 0, the alarm attaches to the adapter manager element. However, alarms with a non-zero NNM object ID sometimes attach to the manager element when the adapter has not yet discovered all the objects in the map.

When the adapter discovers an element, it re-parents the alarms with the newly discovered element. It is possible to specify a particular alarm column or varbind as the event source and affected element from the Event Descriptions property page that is available from the root element or from the alarm property pages.

Events normalize into alarms at startup and in real time. Not all events become alarms. Alarm normalization data is in a file in the `/OperationsCenter_install_path/database` directory on a per-adapter instance. If the file does not exist, a default file is created. The file name consists of concatenating `NNMEventDefinitions` with the adapter name.

Note the following alarm changes that affect both Operations Center and NNM:

- Severity changes are unidirectional between Operations Center and NNM. Changing the severity of an alarm in NNM changes the severity of the corresponding alarm in Operations Center.
- Alarm deletion is bidirectional between Operations Center and NNM. Deleting an alarm in one system deletes the corresponding alarm in the other.

- Category assignments are unidirectional from NNM to Operations Center. Assigning an alarm a category in NNM changes the category of the alarm in Operations Center.
- Alarm acknowledgement and unacknowledgement changes are bidirectional between Operations Center and NNM. Acknowledging (or unacknowledging) an alarm in one system acknowledges (or unacknowledges) the corresponding alarm in the other system. Acknowledgement and Unacknowledgement of alarms does not affect the alarm counts.

Warnings such as the following appear in the `formula.trc` file if objects are no longer in the map (e.g., they have been deleted), but alarms exist for these objects:

```
2007-03-15 12:31:36,371 WARN Integration.nnm.NNM 75 on camaro.DefaultDataSource -
Cache resolver could not resolve object id to associated symbol list for object id
2017; will cache 'null': Object not on map.
```

Loading Historical Alarms

The HP OpenView ovalarmsrv process maintains the current state of the alarms displayed in the native OpenView alarm browsers. However, because ovalarmsrv writes out the state information at periodic intervals, the NNM adapter might be missing historical alarms (at adapter start up) that occurred between the update intervals.

- "Reloading All OpenView Alarms" on page 93
- "Configuring the HP OpenView Ovalarmsrv Process" on page 93

Reloading All OpenView Alarms

To reload all OpenView alarms in Operations Center to restore missing alarm information:

1. In the *Explorer* pane, right-click the top-level server manager element for the NNM adapter, then click *Reload All Alarms*.

   All alarms reload from HP OpenView to the NNM adapter.

Configuring the HP OpenView Ovalarmsrv Process

To configure the HP OpenView `ovalarmsrv` process to write out alarm state more frequently:

1. Use the `-s` argument to specify the interval in seconds in the `ovalarmsrv.lrf` file.

   For example, the following code in the `ovalarmsrv.lrf` file sets the interval to 10 seconds:

   ```
   ovalarmsrv:ovalarmsrv:
   OVs_YES_START:pmd:-s 10:OVs_WELL_BEHAVED:20:PAUSE
   ```

Reducing Delay Time in Reading Events

As described above, setting the recommended alarmsrv parameter to `-s 10` should reduce the delay in reading events when the OpenView adapter starts. However, if a significant delay still exists, define the event reading interval in the */OperationsCenter_install_path*`/config/ovorb.properties` file before starting OvORB for HP OpenView Network Node Manager.

To define the event reading interval:

1. Change the following line in ovorb.properties from:

   ```
   # /opt/OV/bin/ovdumpevents -x 1 -f /tmp/ovdumpevents.out
   ```

to:

```
DumpCmd=/opt/OvORB40/bin/ovorbdumpevents  -f /tmp/ovdumpevents.out -l
<interval minutes> -f1 /tmp/log1
```

This command retrieves the most recent events.

The following code uses a 60 minute interval:

```
# /opt/OV/bin/ovdumpevents -x 1 -f /tmp/ovdumpevents.out
# or otherwise: cmd -f  /tmp/ovdumpevents.out -l 60 > /tmp/log1
DumpCmd=/opt/OvORB40/bin/ovorbdumpevents  -f /tmp/ovdumpevents.out -l 60 -f1 /
tmp/log1
```

2 Start the OvORB.

Permissions on Element Menu Operations

Table 3-17 details the ACL permission on various element menu operations.

Table 3-17 *ACL Permissions for Operations*

Class	Operation	ACL Permission
Root Element	Clear All Alarms	Define
	Reload All Alarms	Define
NNM Element	Manage	Manage
	Unmanage	Manage
	Ping	Access
	Traceroute	Access
NNM Alarm	Acknowledge	Manage
	Unacknowledge	Manage
	Delete	Manage
	Change Severity	Manage
	Change Category	Manage

Troubleshooting

The following are suggestions for resolving common problems that occur in the integration:

- "Operations Center does not reflect status changes" on page 94
- "Submaps are not found" on page 95

Operations Center does not reflect status changes

The map must be read/write to receive status changes from NNM. This is a limitation of the HP NNM API. If the map is not read/write, use the *Refresh* option in the NNM console to update the status.

Submaps are not found

If submaps are not found, perform the following steps:

1. Verify that persistent submaps are on for all levels.

   From the NNM console, click *Map* > *Properties* to open the Map Properties dialog box.
2. Select *IP Map* under *Configurable Applications*, then click *Configure for this Map*.
3. Verify the *On-Demand: to what level should submaps be persistent?* option is set to All Levels.

   By default, this is not automatically set in the Windows version because the Windows version cannot scale as well as the UNIX versions.

The NNM databases that maintain object and topology information can become out of sync. Use the commands listed in Table 3-18, derived from the HP NNM documentation and ITRC forums, to resolve the problem. For more information about these commands, see the ovw reference page in NNM's online help.

Table 3-18 *HP NNM Commands for Synchronizing*

Step	Command	Description
Step 1:	ovstop netmon	Stops the `netmon` service.
Step 2:	ovw -mapcount -ruvDR	Checks the consistency between the map database and the object database. Checks the values of map reference counts stored in the object database and corrects these values, if necessary.
Step 3:	ovtopofix -chs	Runs a utility for cleaning up problems in the topology database and correcting inconsistencies between the topology and the object databases. For more information, see the ovtopofix reference page in NNM's online help (or the TNIX main page).
Step 4:	ovstart netmon	Starts the `netmon` service.

Adding Custom Alarm Properties

It is possible to define custom OpenView alarm properties that are not included in the standard alarm property pages. These custom properties display in the Customer Fields property page. It is necessary to understand how to use the XML-based HierarchyFile, which is a Operations Center mechanism used by adapters to interpret and organize the events reported by management systems.

For more information on the HierarchyFile, see Chapter 9, "Using the HierarchyFile," on page 227.

In order to display custom properties on the Customer Fields property page, it is necessary to define in the HierarchyFile custom alarm fields using an underscore prefix (_). For example: *_TEST*. The Customer Fields property page only displays if there is at least one field defined using an underscore prefix.

The following HierarchyFile displays the Customer Fields property page for all CRITICAL alarms. A custom field named *TEST* displays on this page.

```
<!DOCTYPE hierarchy (View Source for full doctype...)>
  <hierarchy case="yes">
  <!--
 A hierarchy file is required for the NNM Integration.
         The example below can be commented out if not needed.
  -->
  <filter operator="and" invert="false">
  <test type="script" expr="var returnCode = true; var sev = alarm.getField(
'SeverityName' ); if( sev != null ) { if( sev.toString().equals( 'Critical' ) ) {
alarm.setField( '_TEST', 'TEST FIELD' ); } } returnCode;" invert="false" />
  </filter>
  <group name="Alarms" class="gen_folder" affected="no">
  <group name="By Category" class="gen_folder" affected="no">
  <generator field="CategoryName" class="gen_container" affected="yes" hold="no" /
>
  </group>
  </group>
  </hierarchy>
```

3.12.2 HP OpenView Network Node i-series (NNMi)

The NNMi 8 Integration populates Operations Center elements and alarms based on mining information from NNMi 8 inventory objects (Nodes, Node Groups, Interfaces, IP Addresses, IP Subnets, L2 Connections) and incidents.

NNMi 8 incidents and inventory objects are polled by the integration at specific intervals as specified in the *Poll Period (secs)* adapter property. New objects are created when incidents and inventory objects are discovered, existing objects are updated, and objects are removed if no longer returned.

Because the NNMi 8 web services API does not have a last update time for incidents and inventory objects, all objects must be polled for to compare against. If an exception occurs during a poll (i.e. connection timeout) then the remainder of the poll is skipped, and another poll is performed at the regular poll period interval. However, no data should be lost.

The integration is a hybrid of an event-based and object-based integration in the following ways:

- Incident alarms are sent through the hierarchy file to allow for organizing incidents as determined by the hierarchy file.
- By default, alarms representing inventory objects are sent through the hierarchy file to allow for organizing inventory as determined by the hierarchy file. This can be disabled by setting the *Process Inventory Alarms* Adapter property to *false*.
- Elements representing the NNMi inventory objects are represented in the elements tree to reflect the inventory objects as in NNMi.

NOTE: The Operations Center HP NNMi Integration adapter requires a NNMi SDK Enablement license from HP.

Setting up the Integrated Environment

To integrate NNMi:

1. Start the Operations Center server and launch the Operations Center console.
2. Create an HP NNMi Integration adapter for each instance of NNM.

   For instructions, see Section 2.1, “Creating an Adapter,” on page 17.

3 Modify the HP NNMi Integration adapter properties.

For property descriptions, see Section A.14, “HP Network Node Manager i-series,” on page 306.

4 Start the HP NNMi Integration adapter.

For instructions, see Section 2.2, “Starting, Stopping, or Deleting an Adapter,” on page 18.

Customizing the Elements Tree

Each running integration instance has four top level folders: *Incidents*, *Inventory*, *Topology Inventory*, and *Topology Maps*.

- *Incidents* and *Inventory* are event-based branches as determined by the hierarchy file. The names of these two folders and their element structures are determined by the hierarchy file MODL definition.
- *Topology Inventory* and *Topology Maps* are object-based element tree branches.

  Topology Inventory contains an element for each NNMi inventory object (Node, Node Group, Interface, IP Address, IP Subnet, L2 Connection) that is discovered. Only one element for any given NNMi inventory object exists under the *Topology Inventory* element branch, even if that node is a member of more than one node groups.

  Topology Maps element branch represents the hierarchical parent/child structure of Node Groups and Nodes in the same structure as the NNMi web based client’s *Topology Maps/ Node Group Overview* view. There is only one element per inventory object in NNMi. The parent/child relationship is represented in the integration through element links. Child Nodes and Node Groups are linked to parent Node Groups. In addition to showing Nodes and Node Groups, Interfaces are linked to associated Node, and IP Addresses are linked to their associated Interface.

The element tree can be customized by doing the following:

- Change the names of the *Topology Inventory* and *Topology Maps* branches by editing the *Topology Inventory Folder Name* and *Topology Maps Folder Name* adapter properties.
- Create *Topology Inventory* subfolders for inventory objects using the *Use Pattern* adapter property. You can create subfolders if there are a large number of inventory objects. For example, IP Addresses under one parent can be broken out into subfolders representing a portion of the IP Address name, like a subnet.

Property Pages

All incidents and inventory objects have property pages defined but only properties available through the NNMi Web Services API are shown in the adapter. There is an information property page (i.e. “Node Information”, “Interface Information”, etc.) and for inventory objects that support notes, there is a notes property page (i.e. “Node Notes”, “Interface Notes”, etc.) with the notes property.

Operations

Various operations are available for incidents and inventory objects, but only those available via the NNMi Web Services API are shown in the adapter. Since the integration is polling based, changes made in NNMi or through an operation in the integration do not show until the next poll period is complete.

For example, if the integration polls every minute and it takes a minute to poll completely, changing a Node's Note property via the *Change Notes* operation in the integration might not reflect on the Node's property page for two minutes or until the next poll period is complete.

Table 3-19 lists available NNMi operations.

Table 3-19 *NNMi Operations*

Object Type	Available Operations
Incident alarms	◆ Change Lifecycle State. Available settings are Registered, In Progress, Completed, Closed. ◆ Change Notes ◆ Delete
Node Groups and Node Group Inventory Alarms	None
Node Elements and Node Inventory Alarms	◆ Change Management Mode. Available settings are Managed, Not Managed, Inherited, Out of Service. ◆ Change Notes ◆ Delete
Interface Elements and Interface Inventory Alarms	◆ Change Management Mode. Available settings are Managed, Not Managed, Inherited, Out of Service. ◆ Change Notes
IP Address elements and IP Address Inventory Alarms	◆ Change Management Mode. Available settings are Managed, Not Managed, Inherited, Out of Service. ◆ Change Notes
IP Subnet and IP Subnet Inventory Alarms	◆ Change Notes
L2 Connection and L2 Connection Inventory Alarms	◆ Change Notes

Severity and Condition Mapping

In Operations Center, alarms have severity and Elements have condition. The severity of NNMi alarms and condition of elements are based on properties of the NNMi incidents and inventory objects as described in Table 3-20 and Table 3-21.

Table 3-20 *Severity and Condition Mappings for Objects*

Object Type	Property
Incident	severity
Node Group	status
Node	status
Interface	status
IP Address	none
IP Subnet	none
L2 Connection	status

Table 3-21 *Severity and Condition Mapping for Property Values*

Property Value	Severity/Condition
CRITICAL	CRITICAL
WARNING	INFO
MINOR	MINOR
MAJOR	MAJOR
NORMAL	OK
DISABLED	UNKNOWN
UNKNOWN	UNKNOWN
NOSTATUS	UNKNOWN

3.13 HP OpenView Operations for UNIX

The HP OpenView Operations for UNIX Integration (referred to as OpenView) is Operations Center' component for integrating with the HP OpenView Operations product line from Hewlett-Packard.

OpenView supports configurations where the management server can reside on the same or different machine as the Oracle database. The OpenView integration supports both configurations.

The integration relies on efficient sampling of the different database tables for retrieving new and updated message and server data. The OpenView messages contain a number of time stamps that record the message state-transitions that can be extracted for recording business metrics.

Configure the OpenView integration to enable or disable management functions (e.g., acknowledging alarms) within the Operations Center environment. There are three possible integration strategies:

- **OpenView Server Install Integration strategy:** Install Operations Center and configure the HP OpenView Operations for UNIX adapter on the host OpenView server. This configuration enables management functions and does not require installing an ORB on the management server.

- **OVOORB Integration strategy:** Install the OVOORB on each OpenView server. This strategy enables management functions. For more information, see Chapter 10, “ORB Installation,” on page 249.
- **View only Integration strategy:** Install Operations Center and configure the HP OpenView Operations for UNIX adapter. This strategy disables management functions and does not require installing an ORB on the management server.

3.13.1 Integrating Operations Center with HP OpenView Operations for UNIX

Review this section to determine the best strategy for integrating Operations Center with management servers. If there is a default OpenView installation (in a lab for example), proceed directly to the next section, Section 3.13.2, “Quick Start for default OpenView Installations,” on page 101, to begin working with the integration immediately.

To integrate OpenView Operations for UNIX:

1. Define the integration requirements which can include management functions and field values to extract from messages in order to gather business metrics and create element views in Operations Center.

   Use the default OpenView hierarchy XML file shipped with the integration as a template to define customizable views.

2. Identify (or create) the OpenView user for the integration.

   Either define access to nodes and messages based on the user’s responsibility matrix, and let Operations Center inherit these responsibilities, or use the integration in an enterprise-wide configuration which allows access to all of the nodes and messages.

   Most implementations create an OpenView user named Formula and assign some or all of the responsibilities to this user based on the implementation-specific requirements.

3. Optionally install the OVOORB software on each OpenView server.

   For more instructions, see Chapter 10, “ORB Installation,” on page 249.

4. Create an adapter for each instance of HP OpenView Operations.

   For more instructions, see Section 2.1, “Creating an Adapter,” on page 17.

5. Modify the HP OpenView adapter properties.

   For property descriptions, see Section A.15, “HP OpenView Operations for UNIX,” on page 308.

3.13.2 Quick Start for default OpenView Installations

The integration adapter properties default values match the default OpenView Installation.

To integrate OpenView using the default OpenView installation:

1. Create an adapter for the HP OpenView Operations system.

   For more instructions, see Section 2.1, “Creating an Adapter,” on page 17.
2. Set the DB Host adapter property equal to the hostname of the Oracle database server.
3. Set the DB Pass adapter property equal to the password of the Oracle user specified when configuring the adapter.

   The default adapter property value is `OpC_op`.
4. Start the adapter.

3.13.3 View-only Integrations

To create a view-only integration:

1. Create an adapter for the HP OpenView Operations system.

   For instructions, see Section 2.1, “Creating an Adapter,” on page 17.
2. Verify that the following properties are specified: DB Host, DB Name, DB Poll, DB Port, DB Type, and DB User.
3. Leave the following properties unspecified (empty): OVO Password and OVO Server.
4. Start the adapter.

3.13.4 Hierarchy File and Alarm Fields

The integration ships with a default hierarchy file, `DefaultOVOHierarchy.xml`. The file contains a list of all of the available alarm fields that the Operations Center XML generator can evaluate.

3.13.5 Historical Alarms

Historical alarms are acknowledged alarms that exist on the OpenView server.

To display only real-time alarms and hide all acknowledged alarms in Operations Center:

1. Create an alarm filter in the Operations Center console to display only alarms that are not acknowledged (where Acknowledged equals False).

   For more information see *Filtering Alarms* the *Operations Center 5.5 User Guide*.

3.13.6 Alarm Counts

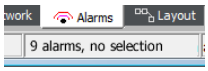
The following status bars in the Operations Center *Alarms* view display alarm counts:

- The filter bar displays the total number of active alarms by severity regardless of alarm ownership:

- The title bar identifies the total number of active alarms by severity with regard to alarm ownership:

  These counts do not include alarms owned by the current and other users.

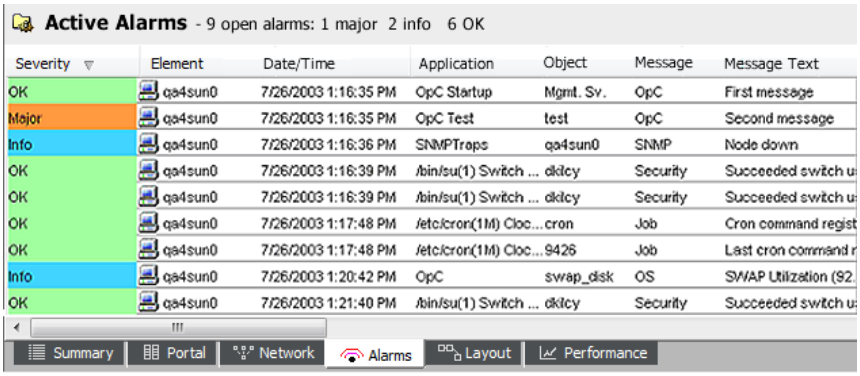
- The status bar identifies the total number of active alarms regardless of alarm ownership.



  It is located at the bottom of the window and shows a single tally of all alarms.

For example, assume nine active alarms are the responsibility of an OpenView user. There are no messages owned by this or any other user. Figure 3-8 shows the Operations Center *Alarms* view for these nine alarms:

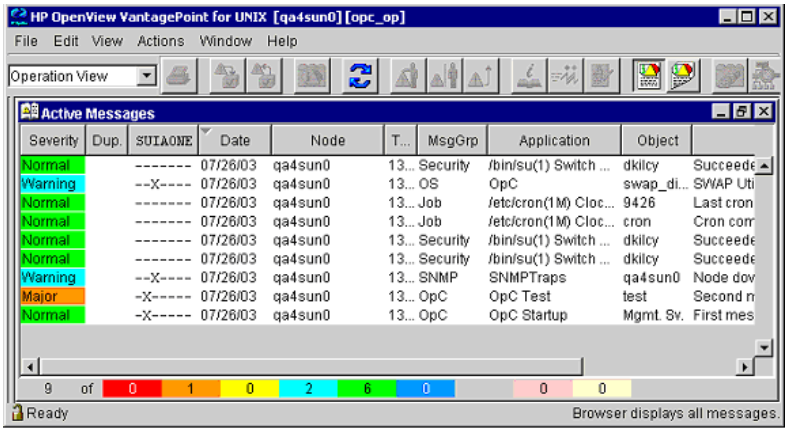
Figure 3-8 *Alarms View*



Both the Filter and Title Bar show 1 MAJOR alarm, 2 INFO alarms and 6 OK alarms. The Status bar shows the total number of alarms equals 9.

Figure 3-9 shows the alarm counts from OpenView's Message Browser. It shows 1 MAJOR alarm, 2 INFO alarms and 6 Normal (OK) alarms, the same as Operations Center. Note the 2 ownership counts (the pink and white counters) both show 0.

Figure 3-9 *OpenVieew Message Browser shows the same alarms as Operations Center*



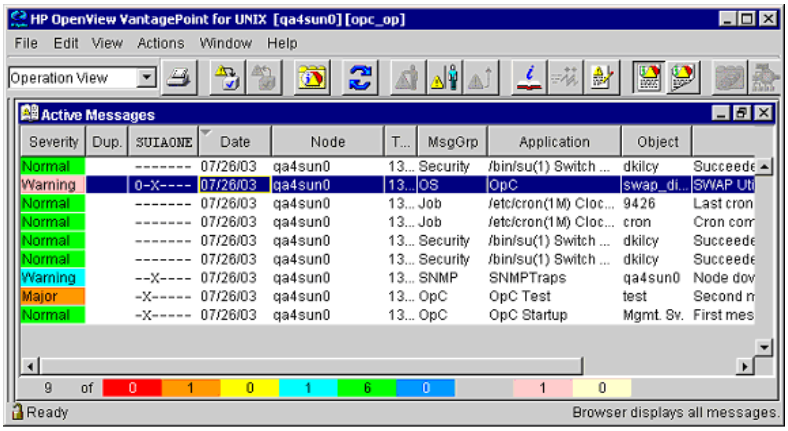
Now, assume the operator right-clicks the INFO alarm with the message SWAP Utilization (92.00%) is greater than 80.00% and then clicks *Own* from the drop-down menu. The operator claims ownership of the alarm. The alarm remains in the browser. The new counts are shown in Figure 3-10.

In Figure 3-9, the filter bar shows 1 MAJOR alarm, 2 INFO alarms and 6 OK alarms. The status bar also shows the total number of alarms, 9 (1 + 2 + 6). Neither indicator changes based on alarm ownership.

However, note that the title bar count changed. It now shows 1 MAJOR alarm, 1 INFO alarm, and 6 OK alarms. Alarm ownership affects these counts. When the operator claimed ownership of the alarm, the number of INFO alarms in the title bar count decreased by 1. Note that the alarm is still visible in the *Alarms* view.

In Figure 3-10, OpenView's message browser alarm counts show 1 MAJOR alarm, 1 INFO alarm, and 6 OK alarms. Note the "owned by me" counter (the pink one) increased by 1.

Figure 3-10 *OpenView Message Browser: The pink "owned by me" counter increased after the alarm ownership changed*



To summarize:

- Alarm ownership affects the title bar counts.
- The filter and status bars in Operations Center display the total number of alarms, regardless of alarm ownership.
- Operations Center does not display the "owned by me" and "owned by others" counter displayed in the OpenView message browser window.

3.13.7 Alarm Operations

The following operations are available for all alarms:

- **Properties:** Shows all the details about a single message.
- **Print:** Sends output to printer.
- Additional management right-click alarm operations are available to the user depending on the integration strategy selected when integrating HP OpenView Operations for UNIX (see "Adding Custom Alarm Properties" on page 95).

To enable these additional management operations, one of the following must occur:

- An existing Operations Center server installation is on the same machine as the OpenView management server.
- The OVOORB must be deployed.

To configure Operations Center on the OpenView Management Server, do the following as necessary:

- "Configuring Right-Click Alarms on Solaris" on page 104
- "Configuring Right-Click Alarms on HP-UX" on page 104

Configuring Right-Click Alarms on Solaris

To configure right-click alarms on Solaris:

1 Add */OperationsCenter_install_path*/lib/libopc.so to the LD_LIBRARY_PATH environment variable on the management server machine.

Configuring Right-Click Alarms on HP-UX

To configure right-click alarms on HP-UX:

1 Add */OperationsCenter_install_path*/lib/libopc.sl to the SHLIB_PATH environment variable on the management server machine.

3.13.8 Management Operations

If configured, additional management operations listed in Table 3-22 are available to users. The OpenView administrator must grant permissions for these operations to OpenView users. The administrator must assign the Manage permission to the user.

Table 3-22 *HP OpenView Operations for UNIX – Alarm Operations*

Operation	Description
Acknowledge	Acknowledges an active alarm. Available if the alarm is unacknowledged.
Delete	Removes an inactive alarm. Available if the alarm is acknowledged.
Add Annotation	Adds an annotation to an active alarm. The annotation feature enables documenting actions and reading through actions that were already performed.
Escalate	Forwards the active alarm (message) to the escalation server. Message escalation must be configured by the OpenView administrator.
Own	Claims ownership of the message. Available if the alarm is not owned by a user.
Disown	Disowns the message, if the message is currently owned by the assigned user.
Start Operator Action	User-initiated action to resolve an event that triggered a message. If an operator-initiated action has been configured for the message, the operation is available only if an action is defined in the OpenView message template.
Start Automatic Action	Performs an automatic action by the management server. If an automatic action has been configured for this message, the operation is displayed.

To access these operations, right-click an alarm and select the operation in the *Alarms* view.

3.13.9 Executing OVO Applications

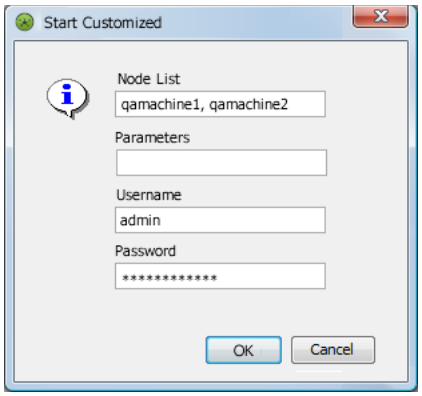
The HP Operations for UNIX adapter provides right-click options for the execution of OVO applications. Applications can be run against the element node or directly from the application element.

- "Execute an OVO Application Against Multiple Nodes" on page 105
- "Executing an OVO Application from an Element Node" on page 105

Execute an OVO Application Against Multiple Nodes

To execute an OVO application against multiple nodes:

1. In the *Explorer* pane, expand *Elements*, then select an OVO adapter.
2. Right-click an application element, then click *Start Customized* to open the Start Customized dialog box:



3. In the *Node List* field, enter element nodes.

   Separate each entry with a comma.
4. In the *Parameters* field, specify any program arguments or options.
5. In the *Username* and *Password* fields, enter the user name and password.
6. Click *OK* to execute the selected application.

Executing an OVO Application from an Element Node

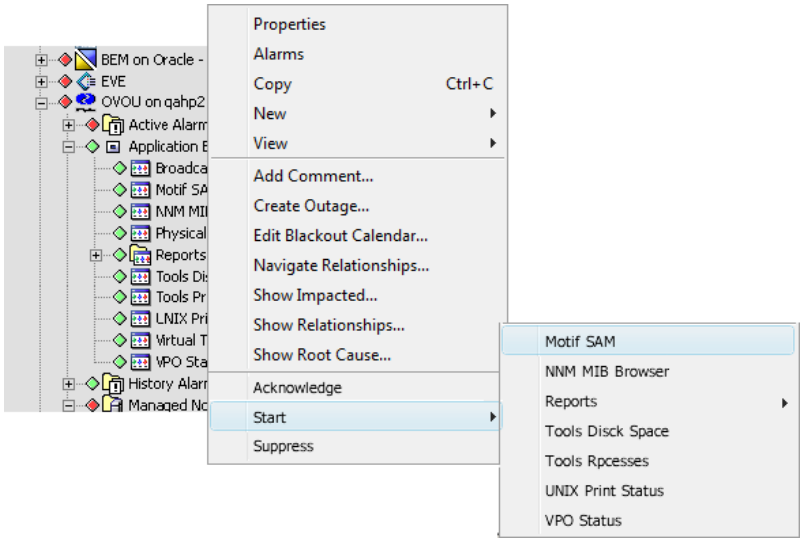
To execute an OVO application from an element node:

1. In the *Explorer* pane, expand *Elements*, then select an OVO adapter.
2. Right-click an element.

3 Click *Start*, then click an application.

The application is executed against the associated element node.

Available options vary depending upon the applications set up in the OVO management system.



3.14 IBM Micromuse Netcool

This product was previously named Micromuse Netcool/OMNIBUS. Create an adapter for each instance of a Netcool object server on the network (see Section 2.1, “Creating an Adapter,” on page 17). Also modify the adapter properties (see Section A.17, “IBM Micromuse Netcool,” on page 311).

If the Netcool user account names do not match the Operations Center user account names, edit the `/OperationsCenter_install_path/database/examples/NetcoolAccountMap.properties` file. The format for mapping the Operations Center user account names to the corresponding Netcool names is:

```
formula01=ncool05
```

Multiple group definitions with the same class or group name cause warning messages to appear in the `formula.trc` file. Duplicate group names do not display in the Operations Center console.

3.14.1 Integration Using a Secure Relay Connection

The Netcool adapter can communicate with the Netcool server using a relay connection to provide secure cross-host communications. The relay connection acts as an intermediary, accepting and delivering messages to one server to another.

All the relay connection components are installed automatically with the Operations Center product. Unzip the */OperationsCenter_install_path*/Relay.zip file and use the following instructions to set up the relay connection between Operations Center and the server:

- "Setting Up the Relay Connection Between Operations Center and the Netcool Server" on page 107
- "Setting Security Parameters" on page 107

Setting Up the Relay Connection Between Operations Center and the Netcool Server

To set up the relay connection between the Operations Center and the Netcool server:

1. Follow the instructions for installing the relay application in the `readme.txt` file on the Operations Center CD.

   This includes information on modifying the `config/relay.properties` file.
2. Configure the Netcool adapter properties that pertain to the relay.

   Note the following port requirements for using the relay connection:

   - The relay requires an open port for relay administration, set in the mosrelay.adminPort property in the `config/relay.properties` file. This port might not be used for any other purpose.
   - In addition, configure each relay with its own port for each required listener configuration. Refer to the `config/relay.properties` file for information on configuring relay listeners.

   Each relay is defined by a set of properties prefixed with "mosrelay.<RelayName>", where <RelayName> is the name of the relay. Each relay definition must contain listener properties specifying how client connections are received. For the required listener properties, see the `config/relay.properties` file.

Setting Security Parameters

The mosrelay.<RelayName>.listener.security property specifies the security level for connections accepted on the listener port. The three valid values are:

- **ssl:** All communications are encrypted with SSL.
- **sslWithClientAuth:** SSL with client certification authentication.
- **unsecured:** Cleartext communications.

To support SSL, supply a trusted server certificate for the relay. If not using the default keystore in the `/config` directory, use the specified properties in `relay.properties` to point to the appropriate keystore.

Operations Center validates SSL certificate dates and flags certificates with expired dates or dates that are not yet valid.

If using self-signed certificates, the process for creating and trusting certificates is identical to the process for the Operations Center server.

If your relays are configured to verify client certificates, remember that the Operations Center server certificate must also be trusted by the VM running the relay.

On the adapter side, specify the following Netcool adapter properties to transmit communications to/from the Netcool server:

- **RelayServe:** Name of the server on which the relay connection exists.
- **RelayPort:** The port number configured for use by the Netcool adapter for relay communications.
- **RelaySecurity:** The security level for the relay server: SSL or unsecured (meaning use cleartext, which is not case-sensitive).

Note there are three valid values in the `config/relay.properties` file and two valid settings for the RelaySecurity adapter property.

What is important to remember about these two sets of settings: If the adapter is set to SSL security, the relay can be set to SSL or sslWithClientAuth. The names are not case sensitive.

In the case of sslWithClientAuth, the relay requests and validates the Operations Center server certificate as part of the SSL handshake. In either case, the adapter should be set to SSL.

3.14.2 Optimizing Alarm Storage

If the Netcool system contains a large number of alarm columns, but only a few are relevant in Operations Center, configure the Alarm Columns adapter property (see Section A.17, “IBM Micromuse Netcool,” on page 311) to load only the selected alarm columns. Operations Center does not mine any other alarm data. This can conserve memory and alarm history storage space.

3.15 IBM Tivoli Enterprise Console (T/EC)

Operations Center can integrate with IBM Tivoli Enterprise Console without the ORB software by using T/EC’s preexisting networking capabilities. Optionally, use the TecORB to improve performance. In either case, synchronize T/EC with Operations Center to complete the integration. The integration relies on a rule-based integration with T/EC.

To integrate T/EC:

1. (Optional) Install the TecORB.

   For instructions on how to perform the installation, see Chapter 10, “ORB Installation,” on page 249.
2. (Optional) Create a HierarchyFile for T/EC systems.

   For instructions, see Chapter 9, “Using the HierarchyFile,” on page 227.
3. Create a T/EC adapter for each instance of T/EC on the network.

   For instructions, see Section 2.1, “Creating an Adapter,” on page 17.
4. Modify the adapter properties.

   For property descriptions, see Section A.2, “Blade Logic Operations Manager,” on page 281.

3.16 IBM Tivoli Enterprise Console (T/EC)+, Database Edition

Operations Center can integrate with IBM Tivoli Enterprise Console with its T/EC+ database-based integration, which provides better performance while lowering stress and impact on the T/EC Enterprise Server.

This adapter provides three integration strategies, all of which access the database directly to retrieve event history, add events, update events, and delete events. This greatly improves event historical performance and reduces the load placed on the T/EC Enterprise Server. These integration strategies are:

- **DB Integration strategy:** retrieves events directly from the database and pushes event updates directly to the database.
- **TecORB Integration strategy:** retrieves events directly from the database, but pushes event updates to T/EC via TecORB, using the Java implementation of the T/EC wsetemsg utility, which tends to provide better performance depending on the T/EC configuration. The TecORB is required in order to use wsetemsg, to accommodate a situation in which Operations Center is installed on a platform that differs from the T/EC platform.
- **Postemsg Integration strategy:** retrieves events directly from the database and pushes event updates to T/EC using a Java implementation of the T/EC postemsg utility. This strategy is applicable for those who are less concerned about the performance of postemsg and/or prefer not to deploy the TecORB.

All strategies require knowledge of connection characteristics and an account on the database server where T/EC sends its event data. Operations Center logs into this database to obtain event data, and after initial mining occurs, polls this table for updates using an efficient, time stamp-based mechanism. There is no additional overhead used by the integration with the exception of one native T/EC Console logged into the database server.

3.16.1 DB Integration Strategy

To set up the T/EC+ adapter to use the DB Integration Strategy:

1 If running Oracle 8 for the T/EC+ database and Oracle for the Operations Center Configuration Storage database, do the following to avoid driver conflicts:

1a Stop the Operations Center server.

For instructions, see “Stopping the Operations Center server in Windows” and “Stopping the Server and the mosdaemon manually in UNIX” in the *Operations Center 5.5 Server Installation Guide*.

1b Install and configure a remote container.

For more information about installing and configuring a remote container, see Chapter 6, “Using Remote Containers,” on page 197.

1c Save the corresponding version of the following JDBC drivers provided by Oracle in the *`ContainerInstall`*`/classes/ext` directory:

```
ojdbc14.jar
```

`orai18n.jar`

1d Start the Operations Center server.

1e Start the container.

For more information about starting a remote container, see Section 6.4.2, "Starting, Monitoring or Stopping a Remote Container Server from the Command Prompt," on page 205.

1f Continue to Step 4 on page 110 and create the T/EC+ adapter under the container.

For additional considerations about creating an adapter in a remote container, see Section 6.6, "Configuring Adapters on Remote Containers," on page 209.

2 If running Oracle 8 for the T/EC+ database but not using Oracle for the Operations Center Configuration Storage database, save the following drivers provided by Oracle in the `/OperationsCenter_install_path/classes/ext` directory:

`ojdbc14.jar`
`orai18n.jar`

3 Restart the Operations Center server.

For instructions, see "Manually Starting the Operations Center Server" and "Starting the Operations Center Server in UNIX" in the *Operations Center 5.5 Server Installation Guide*.

4 Obtain an account on the T/EC database server and ensure that the account has:

- Select and update permission for the tec_t_evt_rep and tec_t_slots_evt tables.
- Select permission for the tec_t_severity and tec_t_status_event tables.

5 (Optional) Create a HierarchyFile for the T/EC systems.

For instructions, see Chapter 9, "Using the HierarchyFile," on page 227.

6 Create an IBM Tivoli Enterprise Console (T/EC)+, Database Edition adapter for each instance of T/EC on the network.

For instructions, see Section 2.1, "Creating an Adapter," on page 17.

7 Modify the IBM Tivoli Enterprise Console (T/EC)+, Database Edition adapter properties.

Set the *IntegrationStrategy* property to `DB`.

For property descriptions, see Section A.21, "IBM Tivoli Enterprise Console (T/EC)+, Database Edition," on page 322.

3.16.2 ORB Integration Strategy

Setting Up the T/EC+ Adapter to Use the TecORB Integration Strategy

To configure the T/EC+ adapter to use the TecORB integration strategy:

1 Install the TecORB.

For instructions, see Chapter 10, "ORB Installation," on page 249.

2 Determine the port on which the ORB is set up to listen.

The default port is 1576.

3 Determine the event console used by the T/EC+ adapter to identify itself to T/EC.

This value must be a valid T/EC Enterprise client name of the T/EC system being integrated.

For instructions, see the EventConsoleName adapter property in Table A-24 on page 319.

4 Implement T/EC Rulebase modifications.

For instructions, see "Implementing T/EC Rulebase Modifications" on page 111.

Implementing T/EC Rulebase Modifications

To implement T/EC rulebase modifications:

1 Insert the following lines into the EVENT class definition of the `root.baroc` file, where the hostname is the name of the T/EC server:

```
originating_tec_hostname: STRING, default="hostname";
originating_event_id: STRING;
originating_date: STRING;
```

2 Create and save a `formula.rls` file with the following code in the `TEC_RULES` directory:

```
/*
   The following rule forwards events to Formula
*/
rule:
setup_formula_slots_rule:
(
  description:'Forward all alarms to Formula Server',
  event: _event of_class _class
    where [
           date_reception: _date_reception,
           event_handle: _event_handle,
           server_handle: _server_handle
          ],
    reception_action:
    (
      sprintf(_orig_date, '%ld',[_date_reception]),
      bo_set_slotval(_event,originating_date,_orig_date),
      re_mark_as_modified(_event,_)
    ),
    reception_action:
    (
      sprintf(_event_id,
'%d%d%ld',[_event_handle,_server_handle,_date_reception]),
      bo_set_slotval(_event,originating_event_id,_event_id),
re_mark_as_modified(_event,_)
    )
).
```

3 Add the new ruleset to the rulebase by adding the following line as the first line of the `TEC_RULES/rule_sets` file:

```
rule_set( 'formula', 'formula.rls', active ),
```

4 Compile the rulebase by issuing the following command, where *rulebaseName* is a variable:

```
wcomprules -t rulebaseName
```

5 Load the rulebase by issuing the following command, where rulebaseName is a variable:

```
wloadrb rulebaseName
```

6 Stop and start the T/EC Enterprise Console Server by issuing the following commands:

```
wstopesvr
wstartesvr
```

7 (Optional) Create a HierarchyFile for the T/EC systems.

For instructions, see Chapter 9, "Using the HierarchyFile," on page 227.

8 Create an IBM Tivoli Enterprise Console (T/EC)+, Database Edition adapter for each instance of T/EC on the network.

For instructions, see Section 2.1, "Creating an Adapter," on page 17.

9 To modify the IBM Tivoli Enterprise Console (T/EC)+, Database Edition adapter properties, set the IntegrationStrategy property to:

```
ORB,hostname=TecORB-Hostname,port=TecORB-Port,console=T/EC EventConsoleName
```

For property descriptions, see Section A.21, "IBM Tivoli Enterprise Console (T/EC)+, Database Edition," on page 322.

3.16.3 Postemsg Integration Strategy

To set up the T/EC+ adapter to use the Postemsg Integration Strategy, implement the following T/EC Rulebase modifications:

1 Insert the following lines into the EVENT class definition of the `root.baroc` file, where *hostname* is the name of the T/EC Server:

```
originating_tec_hostname: STRING, default="hostname";
originating_event_id: STRING;
originating_date: STRING;
```

2 Insert the following modification to the `tec.baroc` file:

```
TEC_CLASS:
    TEC_Sync ISA EVENT
    DEFINES {
        new_status: STRING;
        formula_user: STRING;
    };
END
```

3 Create and save a `formula.rls` file with the following code in the `TEC_RULES` directory:

```
%%
%% The following rule processes TEC_Sync events from Formula
%%
rule: tec_sync_from_formula: (
        description:'Processes TEC_Sync events from Formula Server',
        event: _event of_class 'TEC_Sync'
        where [
                originating_event_id: _orig_event_id,
                formula_user: _formula_user,
                new_status: _new_status
        ],

        reception_action: (
                first_instance(
                event: _orig_event of_class _class
                where [
                        originating_event_id: equals _orig_event_id,
                        status: _status outside [_new_status]
                ] ),
                set_event_status(_orig_event,_new_status),
                bo_set_slotval(_orig_event, administrator, _formula_user),
                re_mark_as_modified(_orig_event,_),
                drop_received_event,
                commit_set
        ).
```

```
%%
%% The following rule sets some slot values for Formula:
%%
rule: setup_formula_slots_rule: (
        description:'Forward all alarms to Formula Server',
        event: _event of_class _class
        where [
                date_reception: _date_reception,
                event_handle: _event_handle,
                server_handle: _server_handle
        ],
        reception_action: (
                sprintf(_orig_date, '%ld',[_date_reception]),
                bo_set_slotval(_event, originating_date, _orig_date),
                sprintf(_event_id,
'%d%d%ld',[_event_handle,_server_handle,_date_reception]),
                bo_set_slotval(_event,originating_event_id,_event_id),
                re_mark_as_modified(_event,_)
        )
).
```

4 To add the new ruleset to the rulebase, add the following line as the first line in the `TEC_RULES/rule_sets` file:

```
rule_set( 'formula', 'formula.rls', active ),
```

5 To compile the rulebase, enter the following command:

```
wcomprules -t rulebaseName
```

6 To load the rulebase, enter the following command:

```
wloadrb rulebaseName
```

7 To stop and start the T/EC Enterprise Console Server, enter the following commands:

```
wstopesvr
wstartesvr
```

8 To complete the integration process:

8a (Optional) Create a HierarchyFile for the T/EC systems.

For instructions, see Chapter 9, "Using the HierarchyFile," on page 227.

8b Create an IBM Tivoli Enterprise Console (T/EC)+, Database Edition adapter for each instance of T/EC on the network.

For instructions, see Section 2.1, "Creating an Adapter," on page 17.

8c Modify the adapter properties:

- Set the IntegrationStrategy property to:

  ```
  IntegrationStrategy=POSTEMSG,hostname=T/
  EC_EnterpriseServerHostname,port=T/
  EC_EnterpriseServerPort,syncClass=TEC_Sync, userSlotName=formula_user
  ```

- If the T/EC Event server is installed on UNIX, or if it is using the default port of 5529 on a Windows installation, it is not necessary to specify the port the T/EC+ adapter learns the port at runtime.
- Acknowledging or closing an alarm updates the T/EC userSlotName property with the Operations Center User ID. If the argument is not present, then no slot is updated.

For property descriptions, see IBM Tivoli Enterprise Console (T/EC)+, Database Edition.

3.16.4 Enabling the T/EC Rulebase for Alarm Suppression

In order to provide options for suppression on T/EC Alarms, it is necessary to setup the T/EC rulebase for suppression.

To enable suppression for T/EC alarms:

1. Using a text editor, open the `TEC_CLASSES/root.baroc` file for the specific ruleset.
2. In the ENUMERATION STATUS section after the 30 CLOSED entry, type 100 SUPPRESSED.
3. Compile the rulebase using the `wcomprules rulebaseName` command.
4. Stop the T/EC event server using the `wstopesvr` command.
5. Start the T/EC event server using the `wstartesvr` command.

   The tec_t_status_event table updates with the *SUPPRESSED* option. The *Suppress* and *Unsuppress* options are available for T/EC alarms.

3.16.5 Using Table Objects in SQL Statements

To enable the use of Table objects in SQL select statements, add the following property to the `/OperationsCenter_install_path/config/Formula.custom.properties` file:

```
Tecdb.DB.Credentials=tec
```

Table 3-23 lists example T/EC select statements.

Table 3-23 *Example T/EC Select Statements*

Objective	Statement
Obtain severity mappings.	`select code, description from tec_t_severity`
Obtain status descriptions.	`select code, description from tec_t_status_event`
Obtain event slot data, use status for initial mining of events, and last_modified_time for pulling updates every 15 seconds.	`select t1.server_hndl, t1.date_reception, t1.event_hndl, t2.slot_name, t2.short_slot_value, t2.long_slot_value from tec_t_evt_rep t1, tec_t_slots_evt t2 where t1.server_hndl = t2.server_hndl and t1.event_hndl = t2.event_hndl and t1.date_reception = t2.date_reception and t1.status <> ? select t1.server_hndl, t1.date_reception, t1.event_hndl, t2.slot_name, t2.short_slot_value, t2.long_slot_value from tec_t_evt_rep t1, tec_t_slots_evt t2 where t1.server_hndl = t2.server_hndl and t1.event_hndl = t2.event_hndl and t1.date_reception = t2.date_reception and t1.last_modified_time >= ?`
Obtain event data, again use status for the initial mining of event and last_modified_time for pulling updates every 15 seconds.	`select * from tec_t_evt_rep where status <> ? select * from tec_t_evt_rep where last_modified_time >= ?`

3.17 IBM Tivoli NetView

To integrate NetView and Operations Center:

1. Install the NvORB software on each NetView server.

   For instructions, see Chapter 10, “ORB Installation,” on page 249.

2 Create a NetView adapter for each instance of NetView on the network.

   For instructions, see Section 2.1, "Creating an Adapter," on page 17.

3 Modify the adapter properties.

   For property descriptions, see Section A.19, "IBM Tivoli NetView," on page 316.

The following topics cover various aspects of managing NetView alarms:

- Section 3.17.1, "Acknowledging and Unacknowledging Alarms," on page 115
- Section 3.17.2, "Managing and Unmanaging Elements," on page 115
- Section 3.17.3, "Different Alarm Totals Displayed in Operations Center and OVW Console," on page 115

3.17.1 Acknowledging and Unacknowledging Alarms

The Operations Center console *Alarms* view provides a Boolean (True/False) alarm column named *Acknowledgement Status for NetView alarms*. Use this data for tracking purposes, such as to check if an operator is working on an alarm.

NetView alarms in Operations Center have *Acknowledge* and *UnAcknowledge* right-click menu options. These options are available for NetView integrations (Solaris and Windows).

If an alarm's *Acknowledgement Status* column value is False (the default) the *Acknowledge* option is available by right-clicking the alarm. Selecting the *Acknowledge* menu option changes the *Acknowledgement Status* value to True.

When an alarm's *Acknowledgement Status* value is True, the *UnAcknowledge* right-click option is available. Selecting *UnAcknowledge* changes the *Acknowledgement Status* value to False.

The acknowledgement status of alarms is not persistent across adapter cycles or shutdowns. Stopping and restarting the adapter does not retain the previous acknowledgement status of alarms. The acknowledgement status field is just for user tracking purposes. It does not perform any action on the NetView console.

Only users with Manage permission can access the *Acknowledge* and *UnAcknowledge* alarm options.

3.17.2 Managing and Unmanaging Elements

Manage or unmanage objects using the element menu options in the Operations Center console. The `Manage` and `Unmanage` element commands are supported for NetView on UNIX (Solaris platforms). For details on using element menu options, see *Managing and Unmanaging Elements* in the *Operations Center 5.5 User Guide*.

3.17.3 Different Alarm Totals Displayed in Operations Center and OVW Console

The alarm totals displayed in the Operations Center *Alarms* view might differ from the alarm totals displayed in the NetView OVW console. For example, the *Alarms* view might display 386 alarms, while the OVW alarm browser displays 395 alarms. Also, the totals for each condition (Critical, Major, Minor, etc.) might differ between Operations Center and OVW.

The NetView event store (where alarms are persisted) is a finite size. When the OVW alarm browser runs for a long period of time, some alarms are rolled over (pushed out) from the event store, but still display in the OVW alarm browser.

When an adapter starts in Operations Center, it mines the NetView persistent event store for alarms. However, Operations Center cannot mine the alarms that were pushed out from the event store, even if they display in the OVW alarm browser.

3.18 Microsoft Operations Manager (MOM)

Operations Center integrates directly with Microsoft Operations Manager without an ORB.

Operations Center logs in to the MOM database to obtain MOM event data. After the initial data mining, Operations Center polls database tables for updates using an efficient, time stamp-based mechanism. The integration with the database uses no additional overhead, with the exception of using one native MOM console logged into the database server.

3.18.1 Integrating to MOM

Users of Microsoft System Center Operations Manager (SCOM) 2005 should create and configure a MOM adapter. Aside from differences in the database names, the integration process is the same for both products.

To integrate MOM (or SCOM 2005):

1. Obtain an account on the MOM database server and ensure that the account has “select” permissions on the following database views:

   Onepoint.alert and Onepoint.alertlevel (requires “update” permission to resolve alarms)
   Onepoint.computer
   Onepoint.processrule
   Onepoint.resolutionstate
   Onepoint.computerattribute
   Onepoint.computerattributedefinition

   If using SCOM, the default database name is OperationsManager (not Onepoint), but this can be changed. Make sure of the exact database name.
2. Create an adapter for Microsoft Operations Manager (MOM)

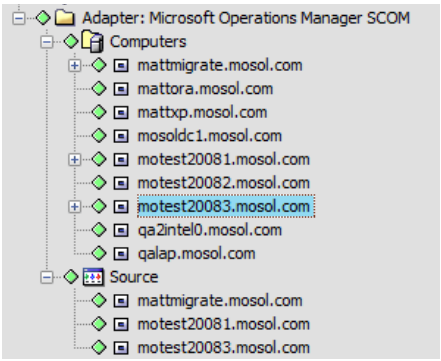
   For instructions, see Section 2.1, “Creating an Adapter,” on page 17.
3. Define the adapter properties.

   For property descriptions, see Section A.23, “Microsoft Operations Manager (MOM),” on page 328.
4. Start the MOM Administrator Console.

3.18.2 Default Hierarchies

By default, alarms brought into Operations Center are grouped by Computer ID or Source.

Figure 3-11 *Explorer Pane: Incoming MOM events are grouped by Computer ID and Source*

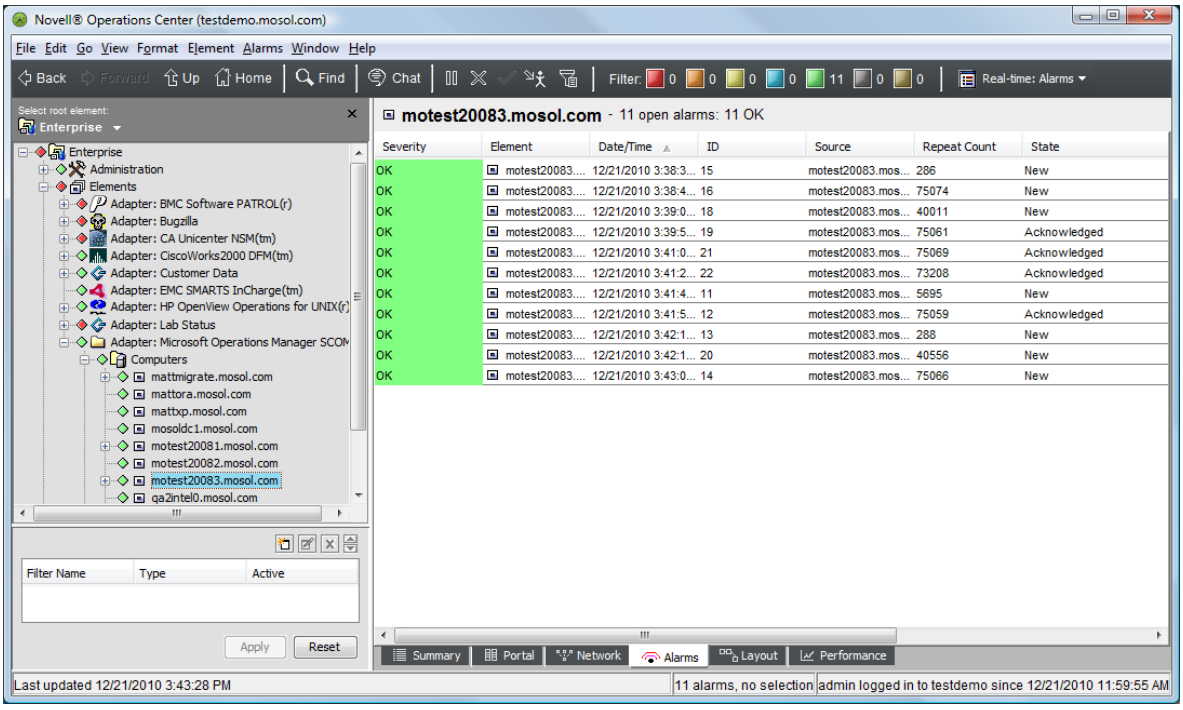


Explorer pane hierarchies can be modified, for example to group events by severity, by modifying the adapter's hierarchy file. For more information on customizing hierarchy files, see Section 2.4, "Customizing the Adapter Hierarchy," on page 19.

3.18.3 Viewing MOM Alarms

Expand the computer names in the *Explorer* pane and identify the source of each alarm, as shown in Figure 3-12:

Figure 3-12 *Operations Center console: MOM alarms grouped by computer ID*



Standard Operations Center columns display for events in the *Alarms* view including: *Severity*, *Element* its name, the *Date/Time* the alarm was received, and the Element's *ID*.

Additional alarm columns are added for MOM adapters including:

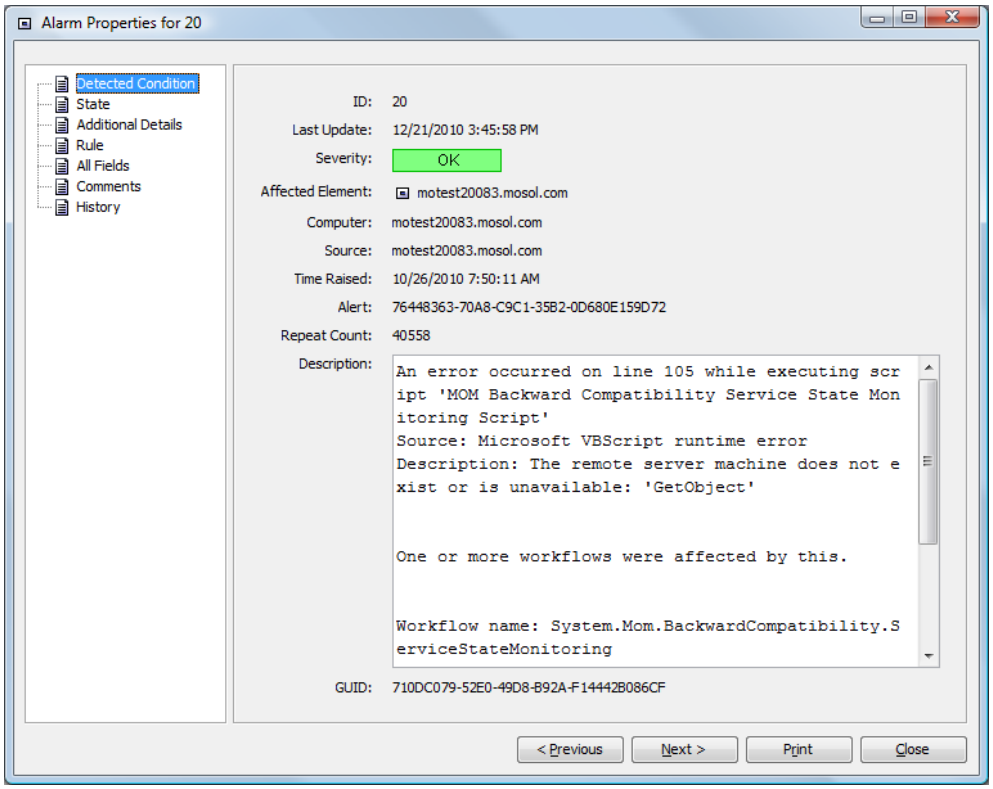
- **Repeat Count:** Number of duplicate identical alarms that this instance represents.
- **Source:** Source of the alarm.
- **State State:** Resolution status of the alarm.

3.18.4 Viewing MOM Alarm Properties

Many commonly referenced alarm properties from MOM surface in the Operations Center alarm properties.

To access the alarm properties:

1 Right-click the alarm in the *Alarms* view, then click *Properties* to open the property pages:



Many of the fields displayed in the alarm property pages originate directly from the MOM alert properties. Refer to the MOM documentation for definitions of these fields.

The following sections list the exceptions and special notes regarding these fields:

- "Detected Condition" on page 119
- "State" on page 119
- "Additional Details" on page 119
- "Rule" on page 120
- "History" on page 120

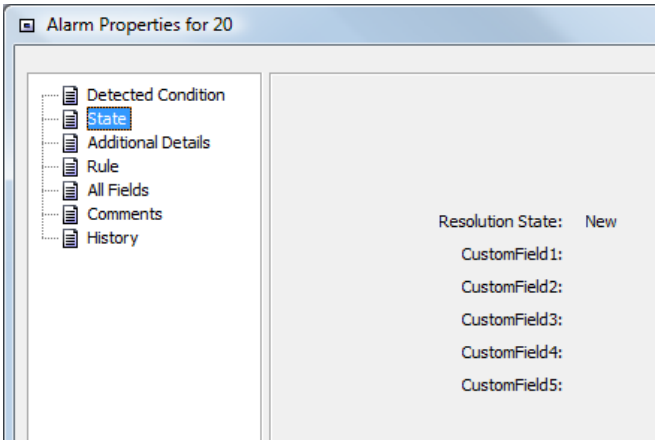
Detected Condition

The Detected Conditions property page corresponds to the *Detected Conditions* tab in MOM. GUID refers to the globally unique identifier for the alarm.

State

The State property page identifies the resolution state of the alarm. The Custom Fields are not operational for the MOM adapter.

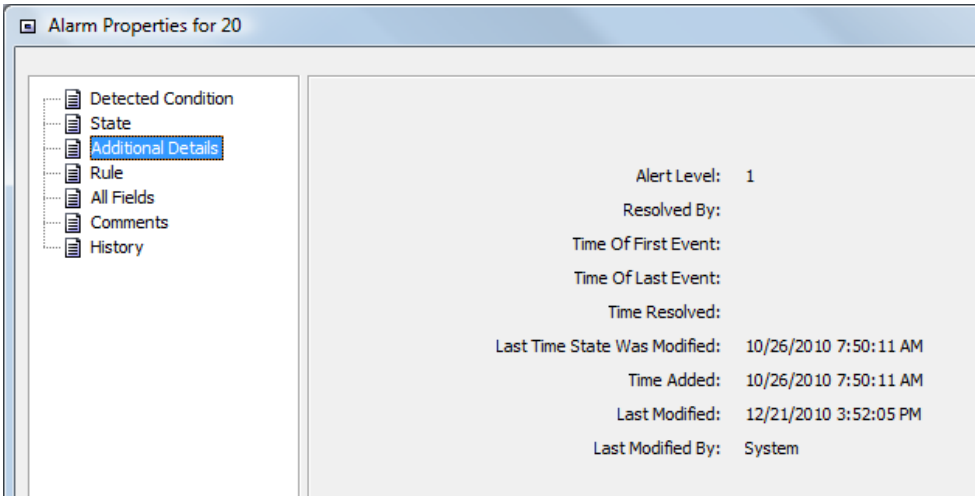
Figure 3-13 *Alarm Properties State Tab*



Additional Details

The Additional Details property page consolidates information from several different areas provides a quick view of the current alarm status – the most recent date and times associated with the last event, the last time the state was updated, and the user who made the most recent modification.

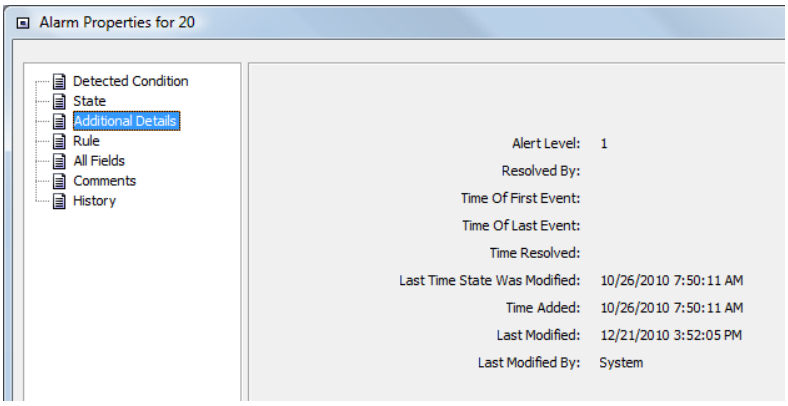
Figure 3-14 *Alarm Properties Additional Details Property Page*



Rule

The Rule property page displays the event processing rule associated with the alarm.

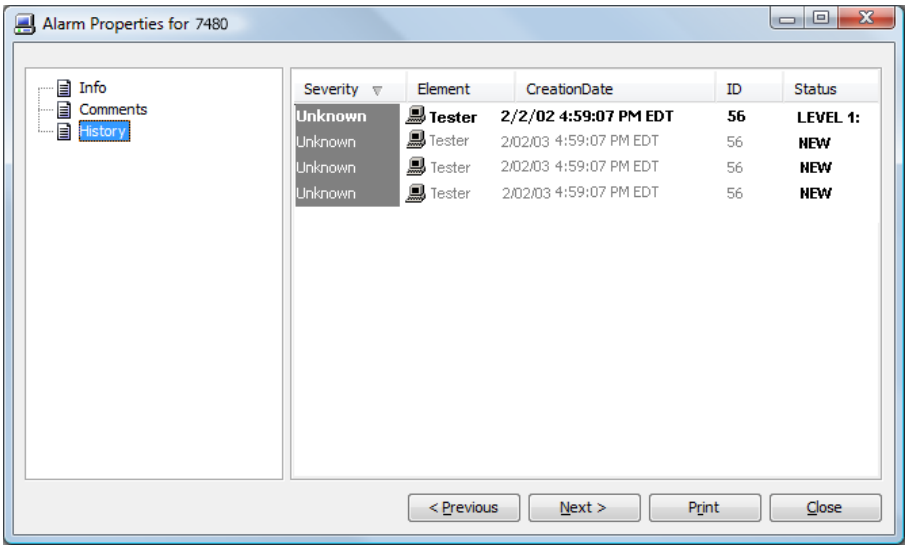
Figure 3-15 *Alarm Properties Rule Tab*



History

The History property page displays the history of alarm-related changes for a particular alarm. Note that the historical information does not originate from MOM resolution history. This information is based only on the Operations Center' alarm history feature and is available only when set up by the administrator.

Figure 3-16 *Alarm Properties History Tab*

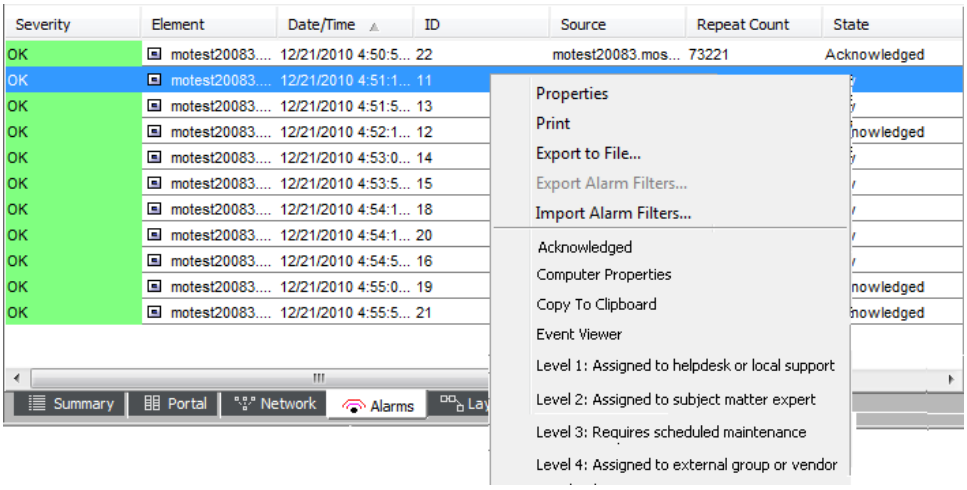


For more information on the Operations Center alarm history feature, see *Capturing Alarm and Performance History* in the *Operations Center 5.5 Server Configuration Guide*.

3.18.5 MOM Alarm Right-Click Options

MOM-related Alarm options as detailed in Table 3-24 on page 121 are available as right-click menu options from MOM alarms.

Figure 3-17 *Example of MOM Alarm Right-Click Menu*

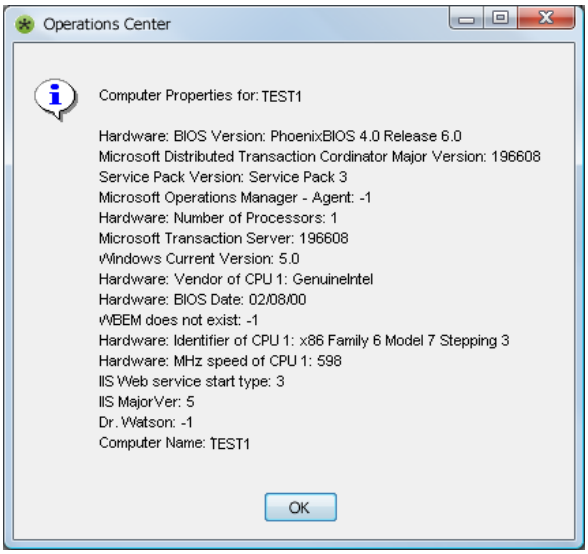


The following options are available on the right-click menu of an alarm associated with MOM:

Table 3-24 *MOM Alarm Right-Click Options*

Option	Function
Properties	Opens the alarm property pages.
Print	Prints the information currently displayed on the screen.
Computer Properties	Displays PC data from MOM database tables, as shown in Figure 3-18.
Copy to Clipboard	Copies the alarm information to the Clipboard. Paste in any application that accepts text.
Event Viewer	Launches the MOM Event Viewer in a separate window.
Process Status Commands	The remaining options change the resolution state of the alarm based on your administrator's configuration in MOM. If these Alert Resolution State settings are changed in MOM, they are available in Operations Center when the adapter is restarted. Refer to the MOM documentation or administrator for the meaning of each status.

Figure 3-18 *Computer Properties Dialog showing PC data from MOM database tables*



3.19 Microsoft System Center Operations Manager (SCOM)

Operations Center integrates directly with Microsoft System Center Operations Manager without requiring installation of an ORB.

NOTE: To integrate with SCOM 2005, it is necessary to use the Microsoft Operations Manager Adapter. For instructions, see Section 3.18, “Microsoft Operations Manager (MOM),” on page 116.

Operations Center uses the Microsoft SCOM SDK to integrate with SCOM, and to retrieve event and/or object hierarchies. Operations Center polls for updates using an efficient, time stamp-based mechanism. The integration with SCOM uses no additional overhead, with the exception of using one SCOM Administrator console account. A Microsoft Windows server is required to integrate with SCOM.

3.19.1 Configuring the Integration

Users of Microsoft System Center Operations Manager (SCOM) 2005 should create and configure a MOM adapter. For instructions, see Section 3.18, “Microsoft Operations Manager (MOM),” on page 116.

If running Operations Center server on a non-Windows server, you must create a Remote Container in order to run the SCOM adapter. For more information about remote containers, see Chapter 6, “Using Remote Containers,” on page 197.

The following sections provide instructions on integrating with SCOM 2007 and SCOM 2012:

Integrating to SCOM

To integrate SCOM:

1 Do one of the following:

1a **SCOM 2007:** Verify that Microsoft.NET 3.5 is installed on the Operations Center server. The Microsoft SDK requires Microsoft.NET 3.5 in order to register files.

- If a later version of Microsoft.NET is installed, verify that version 3.5 was previously installed on the server before upgrading. However, if running Windows 2003, it is necessary to install Microsoft.NET 3.5.
- If running Windows 2008, enable that Role so that it installs.

1b **SCOM 2012 and 2012 SP1:** Verify that Microsoft.NET 4 is installed on the Operations Center server. The Microsoft SDK requires Microsoft.NET 4 in order to register files.

2 Verify that port 5724 on the Operations Center server is open for communications with the SCOM server.

Microsoft(r) has specific port requirements for communications across firewalls. For more information, see the `Supported Firewall Scenarios` topic at Microsoft(r) System Center Operations Manager 2007 Supported Configurations (http://technet.microsoft.com/en-us/library/bb309428.aspx) and Microsoft(r) System Center System Requirements for System Center 2012 - Operations Manager (http://technet.microsoft.com/en-us/library/hh205990.aspx).

3 Do the following to integrate to SCOM 2007:

3a Copy the following files from the `Program Files\` *`System_Center_Operations_Manager_2007_install_path`*`\SDK Binaries\` directory on the System Center Operations Manager (SCOM) server into the `\Windows\assembly` directory.

- `Microsoft.EnterpriseManagement.OperationsManager.Common.dll`
- `Microsoft.EnterpriseManagement.OperationsManager.dll`

It might be necessary to login as `localhost\administrator` to successfully copy these files.

3b Open a command prompt and issue the following commands from the `\`*`OperationsCenter_install_path`*`\Integrations\ext\SCOMIntegration\` directory:

```
C:\Windows\Microsoft.NET\Framework64\v2.0.50727\regAsm.exe /codebase
ScomIntegrationHelper.dll
```

4 Do the following to integrate to SCOM 2012 or 2012 SP1:

4a Copy the following files from the `Program Files\` *`System_Center_Operations_Manager_2012_install_path`*`\Console\SDK Binaries\` directory on the System Center Operations Manager (SCOM) server into the `\Windows\assembly` directory.

- `Microsoft.EnterpriseManagement.Core.dll`
- `Microsoft.EnterpriseManagement.OperationsManager.dll`
- `Microsoft.EnterpriseManagement.Runtime.DLL`

It might be necessary to login as `localhost\administrator` to successfully copy these files.

4b To register the integration dlls, open a command prompt and issue one of the following commands from the `\OperationsCenter_install_path\Integrations\ext\SCOM2012Integration\` directory:

- **SCOM 2012:** `C:\Windows\Microsoft.NET\Framework\v2.0.50727\regAsm.exe /codebase Scom2012IntegrationHelper.dll`
- **SCOM 2012 SP1:**

  `C:\Windows\Microsoft.NET\Framework64\v4.0.30319\regAsm.exe /codebase Scom2012IntegrationHelper.dll`

4c To verify the dll registration, run Regedit (Regedit.exe id stored in the same folder as MS Windows) and search for the following CLSID in the windows registry:

`2C5A508C-0B15-49c3-8059-8D9FE592B65A`

5 Start the Operations Center server.

6 Create an adapter for Microsoft(r) System Center Operations Manager.

From the *Type* drop-down list, select `Microsoft(r) System Center Operations Manager 2007` or `Microsoft(r) System Center Operations Manager 2012`.

For instructions, see Section 2.1, "Creating an Adapter," on page 17.

7 Define the adapter properties.

For property descriptions, see Section A.24, "Microsoft System Center Operations Manager (SCOM)," on page 331.

Upgrading from SCOM 2007 to SCOM 2012

To upgrade from SCOM 2007 to SCOM 2012:

1 Uninstall the following files from the `\Windows\assembly` directory. Click on the file and select *Install*.

- `Microsoft.EnterpriseManagement.OperationsManager.Common.dll`
- `Microsoft.EnterpriseManagement.OperationsManager.dll`

It might be necessary to login as `localhost\administrator` to successfully uninstall these files.

2 Unregister the `ScomIntegrationHelper.dll` file. Open a command prompt and issue the following command from the `\OperationsCenter_install_path\Integrations\ext\SCOMIntegration\` directory:

`C:\Windows\Microsoft.NET\Framework64\v2.0.50727\regAsm.exe /u ScomIntegrationHelper.dll`

3 To verify the dlls were unregistered, run Regedit (Regedit.exe id stored in the same folder as MS Windows) and search for the following CLSID in the windows registry:

`6BFE3547-E221-44A9-9876-D0A2E1906902`

4 Perform Step 4 on page 123 of the SCOM integration procedure.

Upgrading from SCOM 2012 to SCOM 2012 SP1

To upgrade from SCOM 2012 to SCOM 2012 SP1:

1. Unregister the `Scom2012IntegrationHelper.dll` file. Open a command prompt and issue the following command from the `\OperationsCenter_install_path\Integrations\ext\SCOM2012Integration\` directory:

   ```
   C:\Windows\Microsoft.NET\Framework64\v2.0.50727\regAsm.exe /u
   Scom2012IntegrationHelper.dll
   ```

2. Uninstall the following files from the `\Windows\assembly` directory:

   - `Microsoft.EnterpriseManagement.Core.dll`
   - `Microsoft.EnterpriseManagement.OperationsManager.dll`
   - `Microsoft.EnterpriseManagement.Runtime.dll`

   Click each file and select *Uninstall*.

   It might be necessary to login as `localhost\administrator` to successfully uninstall these files.

3. To verify the dlls were unregistered, run Regedit (Regedit.exe id stored in the same folder as MS Windows) and search for the following CLSID in the windows registry:

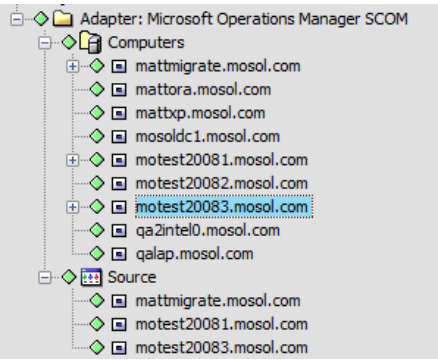
   `6BFE3547-E221-44A9-9876-D0A2E1906902`

4. Perform Step 4 on page 123 of the SCOM integration procedure.

3.19.2 Default Hierarchies

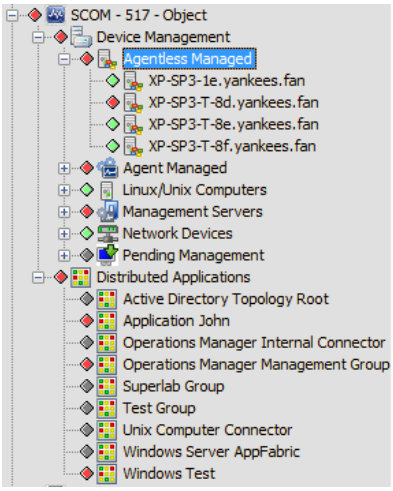
In the SCOM adapter properties, an integration type is defined. Options are to run an Event and/or Object integration. By default, when running an Event integration for SCOM, alarms brought into Operations Center are grouped by Computer ID or Source.

Figure 3-19 *Event Integration: Incoming events are grouped by Computer ID and Source*



However, when running an object integration for SCOM, alarms are grouped based on the SCOM object hierarchy by *Device Management* and *Distributed Applications*.

Figure 3-20 *Object Integration: Incoming events are grouped using the SCOM object hierarchy*



Explorer pane hierarchies can be customized, for example to group events by severity, by modifying the adapter's hierarchy file. For more information on customizing hierarchy files, see Section 2.4, "Customizing the Adapter Hierarchy," on page 19.

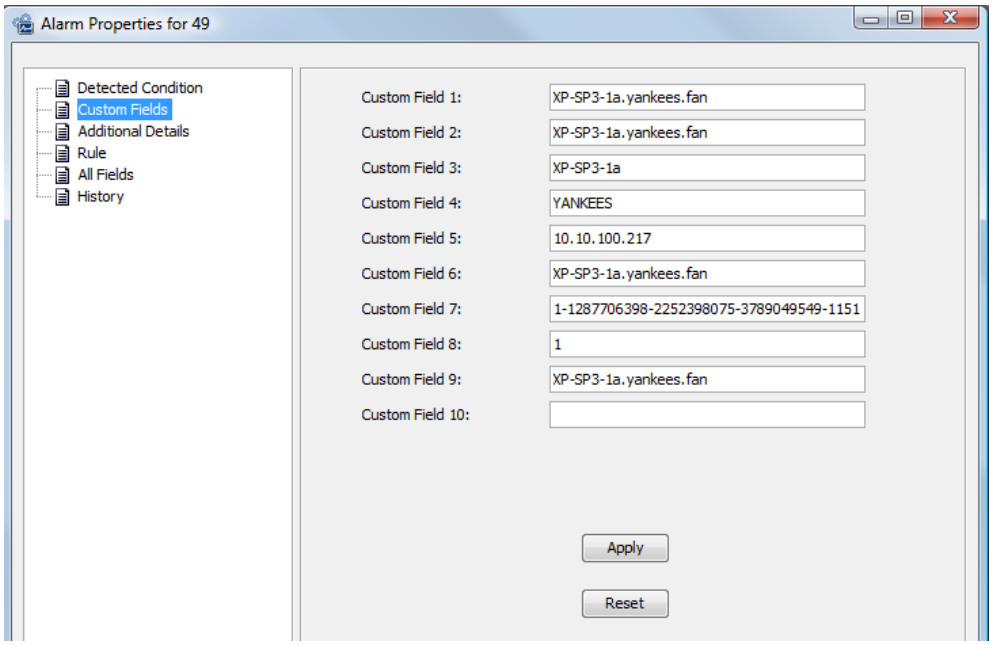
For more information on the Integration Type property, see Section A.24, "Microsoft System Center Operations Manager (SCOM)," on page 331.

3.19.3 Maintaining Custom Fields

The Alarm Properties Custom Fields pane allows you to enter and maintain values for custom fields.

To update custom fields:

1. Right-click the alarm in the *Alarms* view, then click *Properties* to open the property pages.
2. Click *Custom Fields* in the left pane.



3 Update the fields as required and click *Apply*.

3.20 NetIQ AppManager

Operations Center integrates directly with NetIQ AppManager without an ORB. However, the JDBC driver does not support connections using Windows authentication.

If you are using the AppManager Operations Portal, see the *NetIQ AppManager Operations 5.5 Portal Getting Started Guide*.

Refer to the following topics to configure the integration with NetIQ AppManager:

- Section 3.20.1, "Configuring the Integration," on page 127
- Section 3.20.2, "Configuring the NetIQ AppManager Integration for Windows Authentication," on page 128
- Section 3.20.3, "Understanding Element Condition and Alarms in the NetIQ AppManager Integration," on page 128
- Section 3.20.4, "Viewing NetIQ Custom Properties," on page 129
- Section 3.20.5, "Adding NetIQ Alarm Comments," on page 129
- Section 3.20.6, "Using XSL Templates to Transform Event Messages," on page 129

3.20.1 Configuring the Integration

To integrate the NetIQ AppManager:

1 Create an adapter for each instance of a NetIQ object server on the network.

From the *Type* drop-down list, select `NetIQ AppManager`.

For instructions, see Section 2.1, "Creating an Adapter," on page 17.

2 Define the adapter properties.

For property descriptions, see Section A.25, "NetIQ AppManager," on page 332.

3.20.2 Configuring the NetIQ AppManager Integration for Windows Authentication

To configure the NetIQ AppManager adapter to use Windows authentication:

1 Update the *DbName* adapter property to `databaseName;domain=domainName`.

For example, where the DbName value was previously `QDB`, with this change it is now: `QDB;domain=MOS`.

2 Do one of the following to configure the server:

- **On Unix:** No native libraries required, but you must provide user, password.
- **On Windows:** Native libraries are optional. If you don't use the native library, you must provide user, password.
- **On Windows for MSSQL Configuration:** The following requirements must be met:
  - Must be configured to use Windows Authentication.
  - The machine must be on the domain.
  - Users must be a member of the OS's administrator group.

    For example, if you are logging in as MOS\jsmith, the user jsmith must be in the adminitrators group on the MSSQL machine on the MOS domain.
  - If MSSQL is setup to use an instance name, you must turn on the SQLServer Browser Service. Then add the user to the instance's security group.

3 To configure for Windows Single Sign-On (SSO), do the following:

3a If running Operations Center as a service, the service must use the same domain user account as the database.

If you are receiving an `NT AUTHORITY\ANONYMOUS LOGON` exception, verify that the account used by the service and the database are the same.

3b Leave the *User* and *Password* adapter properties blank (empty).

3c Copy the `/OperationsCenter_install_path/classes/win32/ntlmauth32.jar` or `/OperationsCenter_install_path/classes/win64/ntlmauth64.jar` file into the `/OperationsCenter_install_path/classes/ext` directory.

3d Restart the Operations Center server.

For instructions, see "Manually Starting the Operations Center Server" and "Starting the Operations Center Server in UNIX" in the *Operations Center 5.5 Server Installation Guide*.

3.20.3 Understanding Element Condition and Alarms in the NetIQ AppManager Integration

Alarms are handled similar to other event-based adapters in Operations Center. Unacknowledged alarms propagate the adapter's hierarchy at all times.

Element condition in the machine's hierarchy is affected by unacknowledged job alarms, turning the job to CRITICAL and then propagating to the machine element. Acknowledged and closed alarms do not aversely impact the condition of parent elements. If there are no alarms on a job but the job's condition changes for another reason, state is propagated to the *Monitors* element but not to the machine.

3.20.4 Viewing NetIQ Custom Properties

When custom properties have been created for objects in NetIQ AppManager, the same properties are surfaced in the *Object Custom Properties* page in Operations Center properties.

To view object properties:

1. Right-click the NetIQ element in the Operations Center Explorer pane, then select *Properties*. The Properties dialog opens.
2. Click *Object Attributes*, *Object Details*, or *Object Custom Properties* to view object attributes, details or custom properties.

   The pages available for a NetIQ element depend on object type and if any custom properties exist for the object in NetIQ AppManager.

3.20.5 Adding NetIQ Alarm Comments

While Operations Center includes the NetIQ comment feature, there is also an Operations Center-based commenting option:

- "Adding a NetIQ Comment" on page 129
- "Adding a Operations Center-Based Comment" on page 129

Adding a NetIQ Comment

To add a NetIQ comment:

1. Right-click a NetIQ alarm in the Operations Center *Alarms* view, then click *Add AppManager* comment to open the AppManager Comments dialog box.
2. Enter comments, then click *Apply*.

   You can view previous comments in the *Comments History* section.

Adding a Operations Center-Based Comment

To add an Operations Center comment:

1. Right-click a NetIQ alarm in the Operations Center *Alarms* view, then click *Add Comment* to open the Comments dialog box.
2. Enter comments, then click *Apply*.

   You can view previous comments in the *Comments History* section.

3.20.6 Using XSL Templates to Transform Event Messages

NetIQ can contain detailed event messages which are XML documents. Enhancements to the NetIQ integration support XSL templates for transforming these XML documents to HTML. Store the required XSL templates in the `/OperationsCenter_install_path/database/netiq` directory.

- "Integrating the NetIQ XSL Template" on page 130
- "Viewing the NetIQ Event Messages" on page 130

Integrating the NetIQ XSL Template

To create the NetIQ XSL Template:

1 Verify that the NetIQ XML documents have the following format:

```
<?xml version="1.0">
<APPMANAGER>
<DATADETAIL>
...
</DATADETAIL>
</APPMANAGER>
```

The key to the XML document is the first node following the APPMANAGER node, which in this example is DATADETAIL.

2 Place the appropriate XSL template in the */OperationsCenter_install_path*/database/netiq directory.

Use the convention of matching the XSL template file name with the XML document key node. Using the previous example, name the XSL template file `DATADETAIL.xsl`. The adapter uses the XSL template to transform the XML document to HTML.

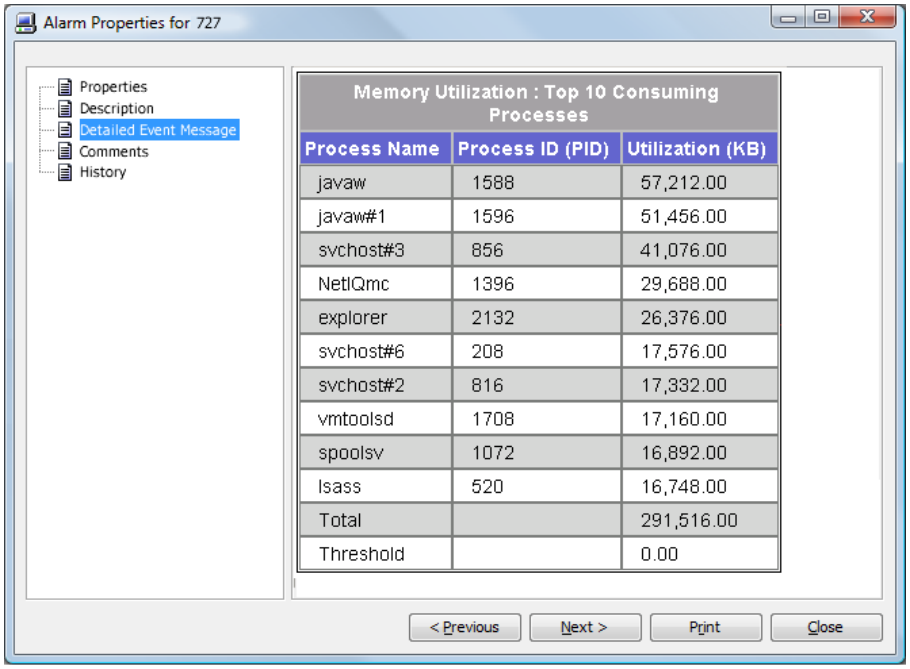
All XSL template files must have the file name extension `.xsl`. The adapter dynamically discovers, loads, and reloads modified XSL template files, so it is not necessary to stop and restart the adapter or the Operations Center server.

Viewing the NetIQ Event Messages

To view NetIQ event messages:

1 In the *Explorer* pane, right-click a NetIQ alarm, then click *Properties* to open the Status property page.

2 In the left pane, click *Detailed Event Message* to open the Detailed Event Message property page:



3.21 NetIQ AppManager Control Center

The NetIQ AppManager Control Center adapter is only used when running the AppManager Operations Portal, and requires that the NetIQ AppManager Operations Portal to be fully implemented. For information on the NetIQ AppManager Operations Portal and the NetIQ AppManager Control Center adapter, see *NetIQ AppManager Operations 5.5 Portal Getting Started Guide*.

3.22 NetIQ Cloud Manager

NetIQ Cloud Manager lets you build a cloud, deliver utility computing, and unify your heterogeneous IT infrastructure with NetIQ cloud management.

Operations Center integrates with NetIQ Cloud Manager to allow you to monitor and manage the following objects:

- Organizations
- Business Services
- Resource Groups
- Workloads

Use the Cloud Manager adapter to:

- Create Service Configurations to link IT infrastructure to your service models.
- Dynamically create Service Level Objectives based on Cloud Manager information.

The following topics describe integrating to NetIQ Cloud Manager:

- Section 3.22.1, "Integrating to NetIQ Cloud Manager," on page 131
- Section 3.22.2, "Implementing a Sample Service Configuration for NetIQ Cloud Manager," on page 132

3.22.1 Integrating to NetIQ Cloud Manager

To integrate NetIQ Cloud Manager:

1. If integrating with a NetIQ Cloud Manager Server configured with SSL, do the following:

   1a. Do one of the following on the NetIQ Cloud Manager server to export the certificate based on your IDS:

   - If using NCSS, issue the following command from the `/opt/netiq/cloudmanager/deploy/ncss/security/nscc/sam/` directory:

     ```
     keytool -export -alias s1as -storepass password -file ncmserver.cer -keystore samKeystore.jks
     ```

   - If using LDAP, issue the following command from the `/opt/netiq/cloudmanager/deploy/ncss/security/nscc/sam/` directory:

     ```
     keytool -export -alias s1as -storepass password -file ncmserver.cer -keystore ncmKeystore.jks
     ```

   Where, *password* is `changeit` by default if not previously modified, or the password for your own CA.

1b On the Operations Center server, do the following to setup authentication with the Cloud Manager server:

1b1 From the location where Java is running, issue the following command to import the ncmserver certificate into the existing cacerts file:

```
keytool -import -keystore cacerts -alias ncmserver -file ncmserver.cer
```

When prompted, specify the password for the certificate or your CA. The default password is `changeit` if not previously modified.

Also be sure to import this into the java cacerts that Operations Center is using. To find out which one is being used, run the Configuration Manager to see what directory it is.

1b2 Issue the following command to view the contents of the certificates and confirm that the Cloud Manager server certificate is added:

```
keytool -list -keystore cacerts
```

1c Restart the Operations Center server.

For instructions, see "Manually Starting the Operations Center Server" and "Starting the Operations Center Server in UNIX" in the *Operations Center 5.5 Server Installation Guide*.

2 Create an NetIQ Cloud Manager adapter.

For instructions, see Section 2.1, "Creating an Adapter," on page 17.

3 Modify the adapter properties.

For property descriptions, see Section A.26, "NetIQ Cloud Manager," on page 334.

3.22.2 Implementing a Sample Service Configuration for NetIQ Cloud Manager

A sample service configuration is provided to help get you started with creating business service views for the Cloud Manager installation. This sample service configuration:

- Builds a service configuration hierarchy for an element under *Services*.
- Links PlateSpin Orchestrate VMs to Cloud Manager workloads under Business Groups.
- Dynamically creates Service Level Agreements and Objectives based on Cloud Manager data by using the `/OperationsCenter_install_path/database/scripts/ncm/setoffer.fs` script. This is configured in the `Scripting` section of the `Modeling Policies` in the service configuration.
- Uses a Join Rule named `Workload to VM` to join workloads in Cloud Manager and VMs within the PlateSpin Orchestrate Adapter. The `VMName` property in Cloud Manager and the `objectId` property in PlateSpin Orchestrate are used in the join rule.

To fully implement the sample service configuration, you'll need to import the configuration and then customize to integrate to the data provided by your adapters.

To setup the sample service configuration:

1. To create a service model, do the following:
   - From the Explorer pane, click to expand *Services*.
   - Right-click *Service Models* and select *Add Element*.
   - Define the service model as required.

   For more information about creating service models, see the *Operations Center 5.5 Service Modeling Guide*.

2. Right-click the new service model, and select *Service Configuration* > *Import*.

   For more information about importing and working with service configurations, see the *Operations Center 5.5 Service Modeling Guide*.

3. Browse to `/OperationsCenter_install_path/database/examples/ncmServiceView.svcconf.xml` and click *Open*.

   A sample service configuration is created under the service model.

4. Right-click the service model, and select *Service Configuration* > *Edit*. The Definition Editor opens.

   For more information about editing service configurations, see the *Operations Center 5.5 Service Modeling Guide*.

5. Do the following to configure the service configuration further:

   - Change the Structure element to point to the NetIQ Cloud Manager Adapter.

     Typically, select Business Groups as the starting point for Structure. In the Definition Navigator, click the *Structures* > *(Element Not Available)* object. Click the icon for *Structure Root* and browse to select *Elements* > Cloud Manager Adapter > *Organizations*. Click *Save*.

   - Change the Source element to point to your PlateSpin Orchestrate Adapter.

     In the Definition Navigator, click the *Sources* > *(Element Not Available)* object. Click the icon for *Structure Root* and browse to select *Elements* > PlateSpin Orchestrate Adapter > *Virtual Machines*. Click *Save*.

6. To run the service configuration, click *Generate*.

   Virtual machines begin to correlate to workloads within your service models. Service Level Objectives are defined for any workloads that have SLOs defined in Cloud Manager. This data can to be surfaced within the dashboard after configuring the associated portlets.

   Service Level Agreement and Objective data can be used by the SLA Compliance Availability Status Report porlet in the dashboard to view the health of your Cloud Manager workloads. For more information on this portlet, see the *Operations Center 5.5 Dashboard Guide*.

3.23 NetIQ Sentinel

The NetIQ (formerly Novell) Sentinel Adapter allows Correlation Rule events to be communicated from Sentinel into Operations Center.

Some additional configurations are necessary to integrate with Sentinel:

- **Sentinel 6:** A Sentinel SMTP Integrator must be configured to send events to the Sentinel Adapter for Operations Center and is used by the Generic Action for Sentinel. If you are already using the existing Sentinel Mail (SMTP) Integrator to send mail, you might need to create a new SMTP Integrator. The Sentinel Generic Event Forwarder is configured to collect all values for a triggered Correlation Rule and forward the values to the Sentinel Adapter.
- **Sentinel 7:** A Sentinel Event Routing Rule and a *Log to Syslog* Action must be configured in order to be able to receive correlated events from the Sentinel Server.

For more information about the Sentinel Generic Event Forwarder, see Sentinel Plug-ins (http://support.novell.com/products/sentinel/secure/sentinel61.html).

To integrate NetIQ Sentinel:

1 To integrate to NetIQ Sentinel 6, do the following:

1a To configure a Sentinel SMTP Integrator, do the following:

1a1 From the *Sentinel Control Center*, click *Tools* and select *Integrator Manager*.

1a2 Select the SMTP Integrator to be used for the Sentinel Adapter.

1a3 Click the Connection panel.

1a4 Specify the Host including IP address of the Managed Objects server and Port used by the Sentinel Adapter to receive events.

1a5 Test and save the configuration.

1b To configure the Generic Event Forwarder, do the following:

1b1 From the Sentinel Control Center, click *Tools* and select *Action Manager*.

1b2 Select Manage Plugins and click the + icon to add the action.

1b3 Select the *Import an Action Plugin File* radio button.

1b4 Click Next.

1b5 Browse to the .zip file for the Generic Event Forwarder and click *Open*.

1b6 Click *Next*.

1b7 Review the information for the action

1b8 Click *Finish*.

1b9 Close the *Action Plugin Manager* panel.

1b10 From the *Action Manager* panel, click the + icon to add another action.

1b11 Specify the name for the new action (ie. Forward Events to Operations Center), and select the *Generic Event Forwarder* from the *Action* drop-down list.

1b12 Select the *Sentinel Mail / SMTP Integrator (configured to connect to the Sentinel Adapter)* from the Integrator drop-down list.

1b13 Select the following values for the action in the *Integrator: Sentinel Mail* column values drop down list:

- **Display Format:** `JSON`
- **Display Data:** `All Data`
- **From Address:** Specify the from address
- **To Address:** Specify the to address.
- **Subject:** Specify the subject line.
- **Display Events:** `All Events`

1b14 Click *Save*.

1b15 Close the *Action Manager* panel.

2 To integrate to NetIQ Sentinel 7, do the following:

2a Download the following files from the JSON-lib Download (http://sourceforge.net/projects/json-lib/files/) Web site, and save them to the */OperationsCenter_install_path/*`integrations/ext/Sentinel` directory:

- json-lib-2.4-jdk15.jar
- json-lib-ext-spring-1.0.2.jar

For more information the JSON-lib, see the JSON-lib (http://json-lib.sourceforge.net/) Web site.

2b Verify the following JSON-lib dependencies are in your classpath;

- jakarta commons-lang 2.5
- jakarta commons-beanutils 1.8.0
- jakarta commons-collections 3.2.1
- jakarta commons-logging 1.1.1
- ezmorph-1.0.6

For more information about dependencies, see the JSON-lib (http://json-lib.sourceforge.net/) Web site.

2c Download the following files from your Sentinel Server at `https://<SENTINEL SERVER IP ADDRESS>:8443/SentinelRESTServices/apidoc/DataObjectAPI.html`, and save them to the */OperationsCenter_install_path*`/integrations/ext/Sentinel` directory:

- sentinel-client-base.jar
- sentinel-client-base-java.jar
- sentinel-client-beans.jar
- sentinel-client-wfbeans.jar

2d To configure Sentinel to log to the Operations Center Sentinel Adapter via Syslog, do the following:

2d1 From the *Sentinel Control Center*, click *Configuration* and select *Integration Manager*.

2d2 Under *Integrators*, select *Syslog*.

2d3 Select *Server Configuration* tab in the right panel.

2d4 Enter the IP address of the Operations Center server used by the Sentinel adapter to receive events in the *Host* field.

2d5 Select *Protocol* from the *TCP* drop-down list.

2d6 Specify the port number in the *Port* field.

This must be the same port as specified in the *Listener Port* property for the Sentinel 7 Adapter in Operations Center.

2d7 Click *Save*.

2d8 From the *Sentinel Control Center*, click *Configuration* and select *Action Manager*.

2d9 Select *Log to Syslog* and click *View/Edit*.

2d10 Select *Event Forwarder* in the *Action* drop down list.

Then, define the following values in the *Action Plug-in To Execute* list:

- **Integrator:** Syslog
- **Display Format:** JSON
- **Display Data:** All Data
- **Display Events:** All Events

2d11 Click *Save*.

2d12 Close the *Action Manager* panel.

2e To configure a Sentinel Event Routing Rule do the following:

2e1 Open the Sentinel Web console by entering the following URL in a Web browser:

`https://SentinelServerAddress:PortNumber`

2e2 On the toolbar, click *Routing*. The Event Routing tab opens.

2e3 Click *Create*.

2e4 Specify the name of the routing rule in the *Name* field.

For example, `All Correlation Events`.

2e5 Enter `st:C` in the *Filter* field.

2e6 Verify the `All` radio button is selected for the *Route the following services* option.

2e7 Select `Log to Syslog` in the *Perform the following actions* drop-down list.

2e8 Verify that the information for the TCP Syslog Server Connection and Port, configured in Step 2d, are correct.

2e9 Click *Save*.

2e10 Verify that *New Event Routing Rule* is selected for the Enabled setting.

3 Create an adapter for each instance of a NetIQ Sentinel on the network. To integrate to Sentinel 7, select `NetIQ Sentinel 7` for the adapter type. For Sentinel 6, select `NetIQ Sentinel` for the adapter type.

For information on creating an adapter, see Section 2.1, "Creating an Adapter," on page 17.

For property descriptions, see Section A.27, "NetIQ Sentinel," on page 335.

3.24 NetIQ Operations Center Experience Manager

Experience Manager works with Operations Center to process and gather application and performance metrics from a wide variety of sources. Use BEM to:

- Establish a response time baseline
- Manage response time, synthetic testing, and application performance from a single application

Experience Manager's and Operations Center's Service Management can evaluate the state of business views in real time and overtime to prioritize repairs and identify performance trends.

For details on using creating and configuring the Experience Manager adapter, see the *Operations Center 5.5 Experience Manager Guide*.

3.25 NetIQ Operations Center Event Manager

Operations Center Event Manager does not need an ORB to work with Operations Center. However, it does require configurations in the Operations Center Configuration Manager and an adapter.

For more information about configuring the Operations Center Event Manager, including adapter properties, see the *Operations Center 5.5 Event Manager Guide*.

3.26 NetIQ Operations Center F/X

The F/X (File and XML) adapter provides the capability to distribute file and XML-based data collection, parsing and processing, and alarm generation using F/X Monitors into multiple BSM engines.

F/X adapters can host any number of F/X Monitors. F/X Monitors can deliver alarm content to any number of F/X adapters. Configure F/X Monitors to multiplex alarms or operate a grouping policy where F/X adapter locations are defined in terms of Primary and Backup mode.

The F/X adapter is a MODL-capable adapter that can receive alarms from multiple F/X Monitors. For details on creating, configuring and using the F/X adapter and Monitor, see the *Operations Center 5.5 F/X Adapter Guide*.

3.27 NetIQ Operations Center SNMP Integrator

The SNMP integrator adapter enables polling, gathering and viewing data available on SNMP agents.

For details on configuring the SNMP adapter, as well as the supported SNMP agents and database requirements, see the *Operations Center 5.5 SNMP Integrator Guide*.

3.28 Novell ZENworks

Novell ZENworks provides desktop configuration management, software distribution, remote management and intelligent image management.

The Operations Center's Novell ZENworks adapter supports both Novell ZENworks Configuration Management and Novell ZENworks Linux Management.

3.28.1 Integrating Novell ZENworks

The Novell ZENworks adapter requires direct access to the Novell ZENworks database in order to run queries. If behind a firewall, verify that your database port is open before integrating to Novell ZENworks.

To integrate Novell ZENworks:

1. If necessary, edit the `Formula.custom.properties` file and add the following properties to override default Sybase database properties used by the ZCM integration:
   - **ZenIntegration.custom.sybase.driver:** the java class name of the driver which needs to be in the classpath. The default is `com.sybase.jdbc3.jdbc.SybDriver`.
   - **ZenIntegration.custom.sybase.url:** the jdbc url. Use `<host>`, `<port>` and `<database>` to represent the configured settings which are replaced at runtime with the actual values. The default is

     `jdbc:sybase:Tds:<host>:<port>/<database>`

     For example, jdbc:sybase:Tds:zcmDbHost:zcmDbPort/ZcmDbName
   - **ZenIntegration.custom.sybase.timequery:** the sql query to retrieve the current time. The default is `select getdate()`.

   For more information about creating or editing the `Formula.custom.properties` file, see "Making Custom Changes" in the *Operations Center 5.5 Server Configuration Guide*
2. Create a Novell ZENworks adapter. For information on creating an adapter, see Section 2.1, "Creating an Adapter," on page 17.

   For property descriptions, see Section A.35, "Novell ZENworks," on page 341.
3. Continue to Section 3.28.2, "Zone Connections," on page 138.

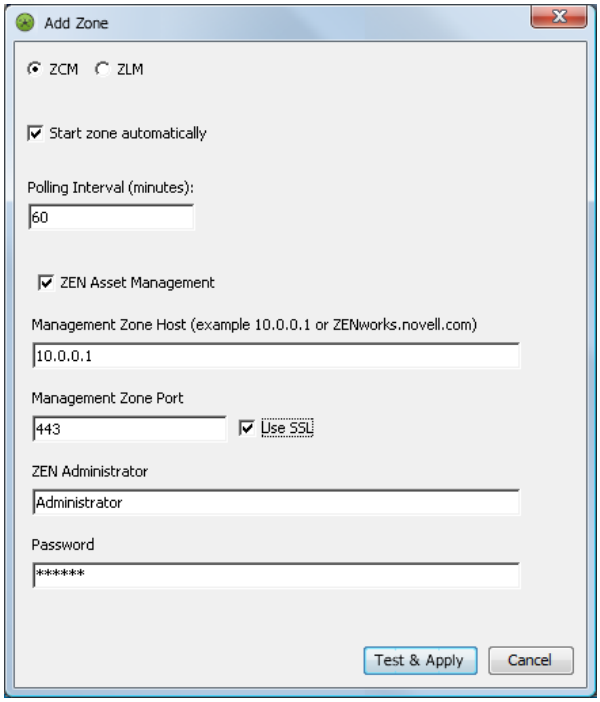
3.28.2 Zone Connections

The following sections provide details about connecting to ZENworks zones:

Adding Zone Connections

To add a new zone connection:

1. From the Explorer pane, expand *Elements* and click to expand the Novell ZENworks adapter instance.
2. Right-click *Administration* and select *Add Zone Connection*. The *Add Zone* dialog opens.



3. Select *ZCM* if the zone is for Novell ZENworks Configuration Management, or select *ZLM* if the zone is for Novell ZENworks Linux Management.
4. Deselect *Start zone automatically* if you do not wish the zone to start automatically when the adapter starts.
5. Select the time in minutes between each poll in the *Polling Interval* field.
6. Specify the properties based on zone type selected.

Field Name	Applicable For...	Description
Management Zone Name	ZLM	The name of the ZLM zone. This is the zone name as specified in the ZLM instance. Contact your ZLM administrator to find or verify the zone name.
Management Zone Host	ZCM and ZLM	The IP address or DNS name of the ZCM or ZLM Server.
Management Zone Port	ZCM	The port to access the ZCM Server. `80` (or `443` for secure connection). Select *Use SSL* if the connection uses SSL.
ZEN Administrator	ZCM	The ZCM Administrator. Default is `Administrator`.
ZEN Administrator Password	ZCM and ZLM	The ZCM or ZLM Administrator password defined during installation. JDBC connection information can be found in the `/etc/opt/novell/zenworks/hibernate.cfg.xml` or `/etc/opt/novell/zenworks/zlm.conf` files which are found on ZENworks Primary Server but requires special permissions to access.
ZEN Asset Management	ZCM	Select this option to display ZENworks Asset Management data in the adapter, such as Licensed Products.
JDBC Driver	ZLM	Oracle or Postgres (default is `Postgres`).
Database Name	ZLM	The name of the database.
Database Host	ZLM	The IP Address or DNS name of the server hosting the database.
Database Port	ZLM	The port for the database. Default is `5432` for Postgres. For Oracle, specify 1521.
Database Username	ZLM	The account to access the database. Default is `zenadmin`.
Database User Password	ZLM	The password for the database account. This password can be obtained from `/etc/opt/novell/zenworks/serversecret` (requires special permissions).

7 Click *Test & Apply* to test the connection and create the zone connection. The zone connection starts automatically unless this option was disabled.

Configuring Zone Severity Mappings

Each zone uses a set of severity mapping values for alarms which can be modified.

To configure alarm mapping values for a zone:

1. From the explorer pane, expand *Elements* and click to expand the Novell ZENworks adapter instance.
2. Expand *Administration* under the ZENworks adapter.
3. Right-click the desired zone and select *Properties*. The *Properties* dialog opens.
4. Select *Severity Mappings*. The *Severity Mappings* panel displays.
5. To specify severity values, do the following as required by the type of zone:
   - Select *Message Log Alarm* and specify the range of values for each severity type.
   - Select *Missing Device Alarm* and specify the severity type based on the number of days since last contact.
   - Select *Critical Patch Alarm* and specify the range of values for each severity type.
   - Select *Recommended Patch Alarm* and specify the range of values for each severity type.
   - Select *Informational Patch Alarm* and specify the range of values for each severity type.
6. Click *Apply*.

Managing Zone Connections

To manage a zone connection:

1. Expand *Administration* under the ZENworks adapter.
2. To edit zone connection properties (requires Define permissions), do the following:
   - **2a** Right-click the desired zone and select *Properties*. The *Properties* dialog opens.
   - **2b** Select the *Zone properties* page.
   - **2c** Change the desired configurations.
   - **2d** Click *Apply*.
3. To connect to a zone (requires Manage permissions), right-click the desired zone and select *Connect to Zone*. The adapter connects to the zone.
4. To refresh data from a zone connection (requires Manage permissions), Right-click on desired zone and select *Refresh Data from Zone*. The zone data refreshes.
5. To disconnect from a zone connection (requires Manage permissions), right-click the desired zone connection and select *Disconnnect from Zone*. The zone connection is disconnected.
6. To delete a zone connection (requires Define permissions), right-click the desired zone connection and select *Delete Zone Connection*. The zone connection definition is deleted.

3.29 PlateSpin Orchestrate

The PlateSpin Orchestrate adapter is designed to display an overview of PlateSpin resources, provide specific operations on VMs, and provide alarms to alert the user of conditions that might need attention.

The adapter displays resources based on:

- Host Servers (Physical)

- Virtual Machines
- Resource Groups
- Repositories

Relationships between the Host Servers and VM's are displayed within the Operations Center hierarchy. Operations available for VM's include Start, Shutdown, Restart, Pause, Unpause, Suspend and Resume.

Repository information and alarms are available to help the user easily determine if available repository disk space is a concern.

To integrate to PlateSpin Orchestrate:

1 Create a PlateSpin Orchestrate adapter or each instance of a PlateSpin Orchestrate on the network.

For instructions, see Section 2.1, “Creating an Adapter,” on page 17.

For property descriptions, see Section A.36, “PlateSpin Orchestrate,” on page 341.

3.30 SolarWinds Orion

SolarWinds Orion is a fault and network performance management platform. Orion stores collected data in an open, Microsoft SQL Server database which allows remote access by external applications such as Operations Center.

Use the Orion adapter to view basic network topology, alarms, remove nodes and interfaces from the native console, and view maps.

To configure a SolarWinds Orion adapter:

1 The Orion software should be installed on the same Windows server where the IIS (Internet Information Services) Web Server and the SQL Server database are installed.

The Application Performance Module must be licensed and installed on top of Solarwinds in order to show applications in the *SolarWinds Orion* adapter for SolarWinds 9.5.

Refer to the Orion system requirements for supported versions of SQL Server.

2 Create a *SolarWinds Orion* adapter for each instance of a SolarWinds Orion server on the network.

For instructions, see Section 2.1, “Creating an Adapter,” on page 17.

3 Modify the adapter properties. For property descriptions, see Section A.38, “SolarWinds Orion Adapter,” on page 344.

Specify the Microsoft IIS server that provides Web dashboard views on the SolarWinds server. The IIS server must allow remote access, and the account you supply must be able to view maps in SolarWinds Orion.

Note the following regarding applications in the *SolarWinds Orion* adapter:

- *Show Applications* is NOT supported in the *SolarWinds Orion* adapter for SolarWinds 9.1.
- If applications are not showing in the SolarWinds adapter as expected, verify the *Show Applications* property is not disabled.
- Performance charting information is available for each application under *Applications* as well as for each application component. Select an application or application component element in the *Explorer* pane, then go to the *Performance* view and select a property from the *Properties* pane to view the performance chart.

4 Discovery Tool Integrations

Networks can grow, shrink and change often. Sometimes, what seems like a clear picture of a network environment can be incomplete. Discovery and dependency mapping tools allow mining a network to detect network devices, applications, and services, thereby creating a more complete picture of the actual environment and its inner workings. They provide powerful visualization and reassurance that a network view is showing the best and most accurate information about an environment.

Operations Center can work with the discovery tools listed in this section:

For information about supported versions of a specific discovery tool, see the *Operations Center 5.5 Getting Started Guide*.

4.1 IBM Tivoli Application Dependency Discovery Manager (TADDM)

Operations Center can integrate with IBM Tivoli Application Dependency Discovery Manager (TADDM) without the ORB software. Integration requires copying a number of TADDM files to the Operations Center installation directory:

4.1.1 Integrating TADDM

To integrate TADDM:

1 If integrating with a TADDM 7.0.x and prior, copy the following files from the TADDM installation directory (`/sdk/lib`) to the `/OperationsCenter_install_path/classes/ext` directory:

```
api-client.jar
api-dep.jar
api-dl.jar
platform-logger.jar
platform-model.jar
```

2 If integrating with TADDM 7.1.2, copy the following files from the TADDM installation directories to the */OperationsCenter_install_path*/integrations/ext/taddm712 directory.

```
IBM/cmdb/dist/lib/platform-jini.jar
IBM/cmdb/dist/sdk/clientlib/taddm-api-client.jar
IBM/cmdb/dist/sdk/lib/platform-model.jar
```

3 If integrating with TADDM 7.2.0.x, copy the following files from the TADDM installation directories to the */OperationsCenter_install_path*/integrations/ext/taddm712 directory.

```
IBM/cmdb/dist/lib/platform-jini.jar
IBM/cmdb/dist/sdk/lib/platform-model.jar
IBM/cmdb/dist/sdk/lib/taddm-api-client.jar
```

4 If integrating with TADDM 7.2.2.x, copy the following files from the TADDM installation directories to the */OperationsCenter_install_path*/integrations/ext/taddm712 directory.

```
IBM/cmdb/dist/lib/platform-jini.jar
IBM/cmdb/dist/sdk/lib/platform-model.jar
IBM/cmdb/dist/sdk/lib/taddm-api-client.jar
IBM/taddm/dist/lib/oal-topomgr-gui.jar
```

5 Verify the TADDM server is configured in Domain mode.

6 Create a TADDM adapter for each instance of TADDM on the network.

For versions prior to TADDM 7.1.2, select the `IBM Tivoli Application Dependency Discovery Manager` adapter type. For TADDM 7.1.2 and higher, select the `IBM Tivoli Application Dependency Discovery Manager 7.2.2.x` adapter type.

For instructions, see Section 2.1, “Creating an Adapter,” on page 17.

For property descriptions, see Section A.18, “IBM Tivoli Application Dependency Discovery Manager (TADDM),” on page 315.

Note for TADDM 7.2.2.x adapter: the xml tag attribute of `case=no` or `case=yes`, set in the `TADDMHierarchy.xml` file and traditionally used to control case of the resulting display data, is ignored.

For more information about using and modifying hierarchy files, see Chapter 9, “Using the HierarchyFile,” on page 227.

4.1.2 Using TADDMHierarchy.xml to Select Data

Use the `TADDMHierarchy.xml` file to select objects from TADDM and display them in the Operations Center console. If you are unfamiliar with the XML-based HierarchyFile structure used with various adapters, see Chapter 9, “Using the HierarchyFile,” on page 227.

In the hierarchy file, the `<generator>` element is used to dynamically create Operations Center elements from elements discovered in a different management system, such as TADDM. The field attribute compares specified fields in the incoming data with a target value, as a way to filter and select data.

- "Updating TADDM Data" on page 152
- "Displaying Node Properties" on page 153

Select by Data Type

Edit the `TADDMHierarchy.xml` file to select TADDM data by using the field attribute of the `<generator>` tag. The field attribute can represent the following data types found in a TADDM discovery:

- **host:** TADDM hosts (servers).
- **networkdevice:** Network devices.
- **applicationcomponent:** TADDM application components.
- **softwareitem:** TADDM software items.
- **businessapplication:** TADDM business applications.
- **appcompsoftwareitem:** An instance of software that defines the application component. This is the main TADDM software item discovered that is most critical for this application server (can exist only under an "applicationcomponent").

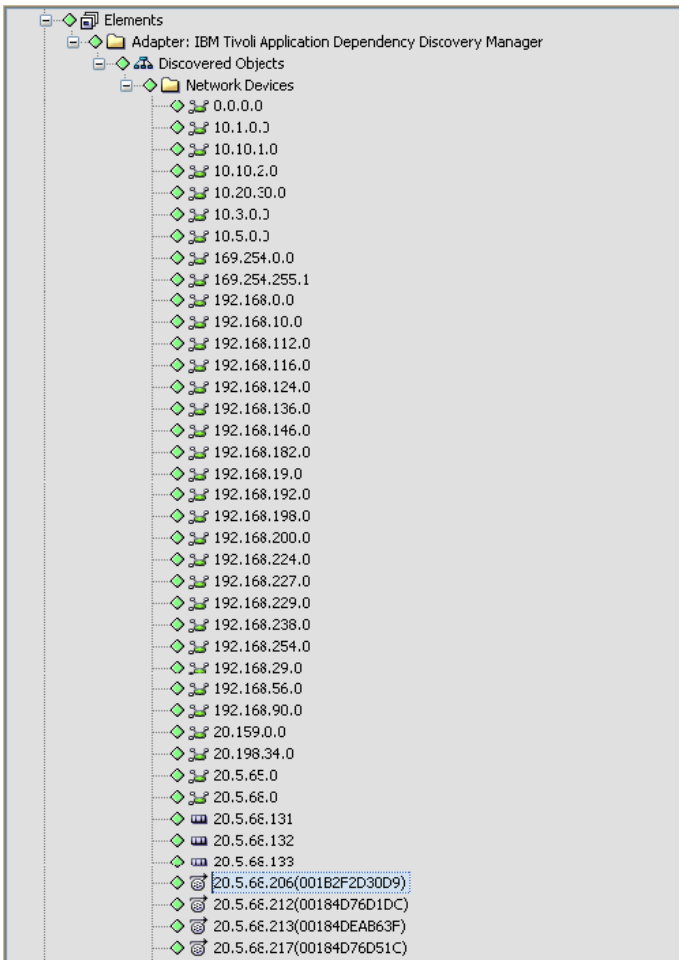
If the class_from_field attribute of the `<generator>` tag is supplied with an icon_name value, then the integration provides a system default icon to represent the generated nodes above.

For example, assume that you want to see all network devices (subnets, switches, etc.) discovered by TADDM. In the integration hierarchy, define the following tags:

```
<group class="gen_folder" name="Network Devices">
  <generator class_from_field="icon_name" field="networkdevice"/>
</group>
```

The resulting view in the Operations Center element hierarchy includes all discovered devices under the Network Devices element:

Figure 4-1 *Operations Center Element Hierarchy*



Select by Fully Qualified Object Names

Refine the data selection process by using TADDM's fully qualified object names to identify objects in a discovery. List the object names as `field` attributes of the `<generator>` tag. Examples:

- com.collation.platform.model.topology.sys.hpux.HpUx
- com.collation.platform.model.topology.sys.UnitaryComputerSystem
- com.collation.platform.model.topology.sys.windows.WindowsComputerSystem

Use Limited Querying

Limited querying can be used to select objects during discovery. An understanding of TADDM's Model Query Language is required to use this feature. For example, set the `field` attribute to `OSRunning.OSName==\'Linux\'*` to select computers that run the Linux operating system. Quotes for the query are escaped with the "\" character.

The structure of OSRunning.OSName follows the tag structure used in the TADDM Model Query Language:

```
SELECT ComputerSystem.displayName FROM ComputerSystem WHERE OSRunning.OSName ==
'Linux'
```

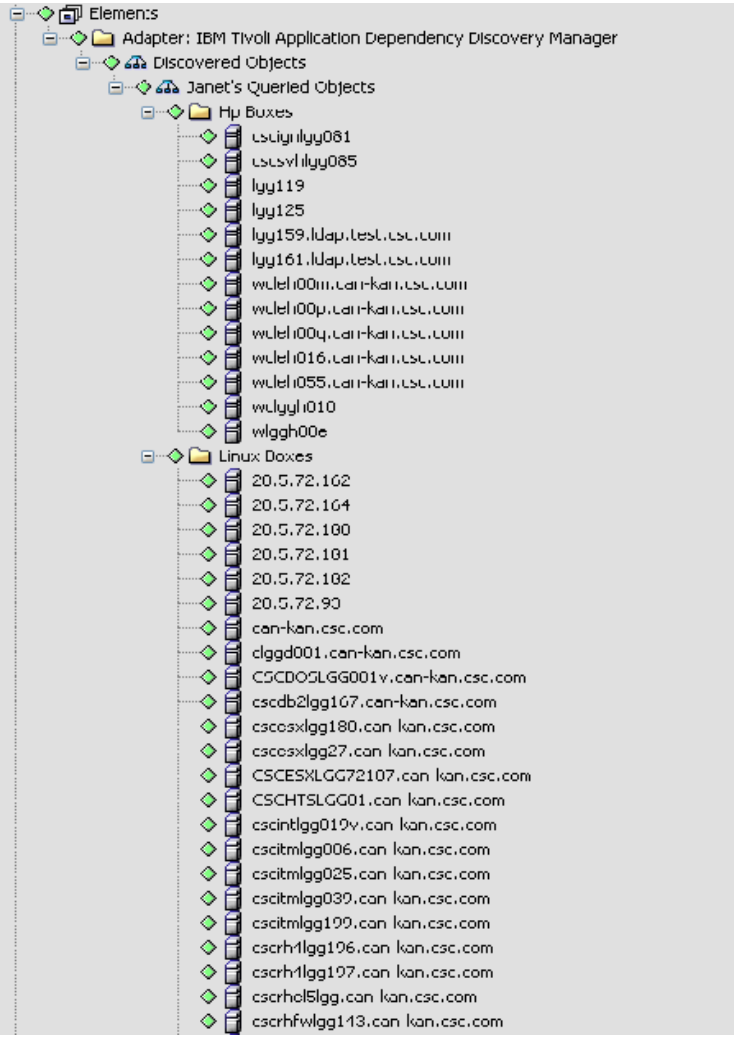
This query returns, as output, an OSName XML tag nested within an OSRunning XML tag, which in turn is nested within a ComputerSystem XML tag.

Assume that you want to see all computers in your network that run HP or Linux. In the integration hierarchy file, define the following, using two different styles of querying in each case:

```
<group class="formula_organizations" name="Janet's Queried Objects">
  <group class="gen_folder" name="Hp Boxes">
    <generator class="admin_automation"
field="com.collation.platform.model.topology.sys.hpux.HpUxUnitaryComputerSystem" /
>
  </group>
  <group class="gen_folder" name="Linux Boxes">
    <generator class="admin_automation" field="OSRunning.OSName==\'Linux\'" />
  </group>
</group>
```

The resulting view in Operations Center shows the computer IDs beneath two parent elements: HP Boxes and Linux Boxes:

Figure 4-2 *Operations Center Hierarchy View*



The TADDM integration can display two types of relationships in Operations Center: explicit relationships and implicit relationships.

Explicit Relationships

Explicit relationships show child data nodes grouped logically beneath parent nodes. These relationships are created by editing the TADDM hierarchy file and nesting `<generator>` tags of a specific field type.

Table 4-1 contains the valid nesting sequences of data for the hierarchy file (the data in the rows represent the field attribute values of the nested `<generator>` tags):

Table 4-1 *Valid Nesting Sequences for Hierarchy File*

Generator Tag field Attribute	Child Generator Tag field Attribute	Grandchild Generator Tag field Attribute
businessapplication	host	networkdevice (a subnet)
businessapplication	host	softwareitem
businessapplication	host	appcompsoftwareitem
businessapplication	applicationcomponent	appcompsoftwareitem
businessapplication	applicationcomponent	networkdevice (a subnet)
businessapplication	host	
businessapplication	applicationcomponent	
host	networkdevice (a subnet)	
host	softwareitem	
host	appcompsoftwareitem	
softwareitem	host	
applicationcomponent	appcompsoftwareitem	
networkdevice	host	

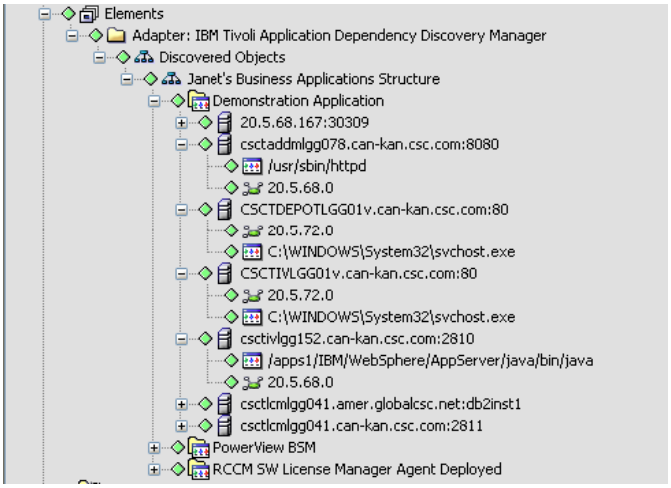
For example, assume that you want to show the Application Components that are placed under Business Applications (i.e., a defined logical structure for grouping discovered objects in TADDM). You also want to show the main software that runs under the Application Component, as well as the subnets to which the server belongs.

In the integration hierarchy file, define the following tags:

```
<group class="formula_organizations" name="Janet's Business Applications
Structure">
  <generator class_from_field="icon_name" field="businessapplication">
    <generator class_from_field="icon_name" field="applicationcomponent">
      <generator class_from_field="icon_name" field="appcompsoftwareitem"/>
      <generator class_from_field="icon_name" field="networkdevice"/>
    </generator>
  </generator>
</group>
```

A defined logical structure for grouping discovered objects in TADDM:

Figure 4-3 *Defined Logical Structure*



The element hierarchy indicates that the TADDM server has three Business Applications defined (Demonstration Application, PowerView BSM and RCCM SW License Manager Agent Deployed).

Under the Demonstration Application element, the second child is a server named "csctaddmllg078.can-kan.csc.com". This server has an application listening on port 8080. Under this server, an HTTP daemon process is running on the server `/usr/sbin/httpd` on the port 8080, and that the `/usr/sbin/httpd"` server is on the 20.5.68.0 subnet.

Implicit Relationships

Implicit relationships display in the Operations Center Relationship Browser. Implicit relationships can show:

- Machines that communicate with one another
- Software that communicates across machines
- Host-to-subnet relationships

These relationships are generated automatically when the appropriate nested generator tags exist within the hierarchy file. In other cases (unnested tags), the relationships must be explicitly enabled using a `<param>` tag.

The Operations Center Relationship Browser provides a graphic view of relationships among elements as well as navigation and layout customization tools. To open the Relationship browser, right-click an element in the *Explorer* pane, then click *Show Relationships*. For more information on using the Relationship browser, see the *Operations Center 5.5 Service Modeling Guide*.

Table 4-2 shows the mapping between <generator> tags and their field attribute values which automatically produces implicit relationships:

Table 4-2 *Valid Nesting Sequences for Hierarchy File*

Generator Tag field Attribute	Child Generator Tag field Attribute	Grandchild Generator Tag field Attribute	Relationships Mapped
businessapplication	applicationcomponent	appcompsoftwareitem	applicationcomponent to applicationcomponent communication
businessapplication	host	appcompsoftwareitem	appcompsoftwareitem to appcompsoftwareitem communication
applicationcomponent	appcompsoftwareitem		applicationcomponent to applicationcomponent communication
host	appcompsoftwareitem		applicationcomponent to applicationcomponent communication

In cases where generator tags are not nested, use the <param> tag. Nest the <param> tag within a <generator> tag to display the relationships in the Operations Center Relationship Browser.

To enable relationship display, set the <param> tag name attribute to showrelationships, and set the attribute value to True. To disable display, set the attribute value to False (which is the default).

For example, to display relationships between network devices:

```
<group class="gen_folder" name="Network Devices">
  <generator class_from_field="icon_name" field="networkdevice">
    <param name="showrelationships" value="true" />
  </generator>
</group>
```

Table 4-3 shows the <generator> tag field attributes that can be used with the <param> tag's showrelationships name attribute, and the resulting relationships:

Table 4-3 *Mapping Relationships using the <Param> Tab*

Generator Tag field Attribute	Param Tag showrelationships set to True?	Relationships Mapped
networkdevice	Yes	All relationships between network devices
host	Yes	If networkdevice is in the same hierarchy, then display the relationship of host to subnet
applicationcomponent	Yes	The relationship between application components that communicate with one another

Relationships and Scoping of Data in Hierarchy File

The structure of the hierarchy file affects how relationships are mapped between elements displayed in the Relationship Browser. In most cases, elements under the same parent structure of <group> tags are considered to be within the same scope. This means that when relationships are mapped, these elements are mapped to one another.

Consider the following XML tags in the hierarchy file:

```
<group class="formula_organizations" name="All of my Structures">
  <group class="formula_organizations" name="My First Set of Structures">
    <generator class_from_field="icon_name" field="businessapplication">
      <generator class_from_field="icon_name" field="applicationcomponent">
        <generator class_from_field="icon_name" field="appcompsoftwareitem"/>
      </generator>
    </generator>
  </group>

  <group class="formula_organizations" name="My Second Set of Structures">
    <generator class_from_field="icon_name" field="businessapplication">
      <generator class_from_field="icon_name" field="host">
        <generator class_from_field="icon_name" field="appcompsoftwareitem"/>
      </generator>
    </generator>
</group>
</group>
```

The items under `My First Set of Structures` are within the same scope and the items under My Second Set of Structures are under a different scope.

However, in some cases, it is necessary to map items that are in different scopes. This occurs in the case of the field types host and network device. Assume that you want to map subnet relationships by setting a <param> tag to `showrelationships` on the host field attribute of a <generator> tag. If both of these tags are present, then the scope is the parent that they both share.

Consider the following data in the hierarchy file:

```
<group class="formula_organizations" name="Discovered Objects" >
  <group class="gen_folder" name="Network Devices">
    <generator class_from_field="icon_name" field="networkdevice">
    </generator>
  </group>
  <group class="gen_folder" name="Hosts">
    <generator class_from_field="icon_name" field="host">
      <param name="showrelationships" value="true" />
    </generator>
  </group>
</group>
```

In this case, items under Network Devices and Hosts are mapped in the resulting subnet/host relationships displayed in the Relationship Browser. This is because they have the common parent Discovered Objects.

Updating TADDM Data

The TADDM adapter properties (see Section A.18, “IBM Tivoli Application Dependency Discovery Manager (TADDM),” on page 315) allow you to set a data refresh interval to make updates from a discovery. Setting a refresh interval is not recommended because relationships are not updated and deleted nodes are not removed. If a new discovery is made in TADDM, the preferred practice is to restart the TADDM adapter. However, if you do set a refresh interval, the minimum allowed poll time is 15 minutes.

Displaying Node Properties

Each TADDM adapter element that in the Operations Center console has a set of properties which can be displayed by right-clicking the element and selecting *Properties*. In the left pane of the properties window, select TADDM Properties.

4.1.3 Scheduling Updates of Discovery Data

In some cases, a TADDM user might periodically re-run discoveries through the TADDM client on a scheduled basis, potentially causing the discovery data in the Operations Center TADDM integration to become out of date.

A scheduled job to refresh the integration at a specified time can be run to refresh the data. Determine how often this restart of the integration needs to occur and when it needs to take place based on how often a discovery is scheduled to run in TADDM.

To run a job to schedule a restart of TADDM:

1. In the Explorer pane, navigate to *Enterprise* >*Administration* > *Time Management* > *Jobs*.
2. Right-click *Jobs* and select *Create Job*. The *Create Job* dialog opens.
3. Specify the name and description of the job in the fields provided.
4. Set the schedule for the job as appropriate.

   For more information about creating and scheduling jobs, see the *Operations Center 5.5 Server Configuration Guide*.
5. Specify the script to run by doing the following:
   - **5a** Click the *Job Script* tab.
   - **5b** Specify the name for the script. For example, `updatetaddm.fs`.
   - **5c** Paste the following script text into the script editor:

```
////////////////////////////////////////////////////////////////      A
utility to stop and then start a TADDM adapter.
////////////////////////////////////////////////////////////////
var adapters = formula.server.adapters();
// The name of the adapter as appears under the "Adapters" node.
var adapterName = "Adapter: IBM Tivoli Application Dependency Discovery
Manager";

for( var i = 0 ; i < adapters.length ; i++ )
{
   if( adapters[i] != null && adapters[i].key().equals( adapterName ) )
   {
      if( !adapters[i].manageStatus().startsWith( "stop" ) )
      {
         writeln( "Restarting TADDM adapter" );
         // Recycle
         adapters[i].manageStop();  // stop ...
         adapters[i].manageStart(); // then start.
      }
```

```
            break;
        }
    }
```

5d Update the value for the *adapterName* variable with the name of your TADDM adapter as it appears under the *Enterprise* > *Administration* > *Adapters* in the Explorer pane.

6 Select *Enable Job*.

4.2 Mercury Application Mapping

This section lists the basic steps for creating a Mercury Application Mapping adapter and provides links to sections that contain more detailed information.

To integrate Mercury Application Mapping:

1 Create the Mercury Application Mapping adapter in Operations Center.

For instructions, see Section 2.1, “Creating an Adapter,” on page 17.

Specify the XML root and configure other adapter properties. For adapter properties, see Section A.22, “Mercury Application Mapping,” on page 326.

4.3 PlateSpin Recon

PlateSpin Recon is a workload profiling, analysis and planning solution. Inventory and workload utilization statistics are collected in a Microsoft SQL Server database which allows remote access by external applications such as Operations Center.

Use the PlateSpin Recon Adapter to visualize your virtual infrastructure and create an inventory of both physical and virtual machines, allowing you to profile server workloads and evaluate performance. For more information about PlateSpin Recon, go to http://www.novell.com/products/recon/.

PlateSpin Recon collects statistical information on various components of the infrastructure including applications, processes, and/or services; and exposes them as individual schedules to provide better control over the memory footprint. Operations Center logs in to the PlateSpin Recon database to import this data collected over time. Query schedules for each data type are defined in the adapter properties to determine when Operations Center polls the PlateSpin Recon databases for new information.

To integrate Operations Center directly with PlateSpin Recon without an ORB:

1 Create an adapter for each instance of a PlateSpin Recon server on the network.

For instructions, see Section 2.1, “Creating an Adapter,” on page 17.

For property descriptions, see Section A.37, “PlateSpin Recon,” on page 343.

Often with discovery tools it can be difficult to estimate the memory demands when an adapter is integrated. Table 4-4 gives you some idea of the additional memory usage demanded on the Operations Center server based on the number of Recon elements:

Table 4-4 *Number of Recon Elements and Additional Memory Usage*

# of Recon Elements	Importing data for…				Estimated Additional Memory Consumption
	Applications	Logs	Processes	Services	
1,000 (Standard Integration)					229.2 MB
	X				1.0 GB
		X			511.9 MB
			X		378.3 MB
				X	918.0 MB
		X	X		738.1 MB
	X			X	1.6 GB
	X	X			1.2 GB
			X	X	1.0 GB
	X		X		1.0 GB
		X		X	1.3 GB
		X	X	X	1.3 GB
	X		X	X	1.7 GB
	X	X		X	2.0 GB
	X	X	X		1.5 GB
	X	X	X	X	2.1 GB

# of Recon Elements	Importing data for…				Estimated Additional Memory Consumption
	Applications	Logs	Processes	Services	
5,000					1.0 GB
	X				4.7 GB
		X			2.3 GB
			X		1.7 GB
				X	4.3 GB
		X	X		3.5 GB
	X			X	8.1 GB
	X	X			5.9 GB
			X	X	4.8 GB
	X		X		4.8 GB
		X		X	6.6 GB
		X	X	X	6.4 GB
	X		X	X	8.3 GB
	X	X		X	9.9 GB
	X	X	X		7.4 GB
	X	X	X	X	10.3 GB

# of Recon Elements	Importing data for…				Estimated Additional Memory Consumption
	Applications	Logs	Processes	Services	
10,000					1.9 GB
(Maximum Integration)	X				9.3 GB
		X			4.7 GB
			X		3.4 GB
				X	8.6 GB
		X	X		6.9 GB
	X			X	16.2 GB
	X	X			11.7 GB
			X	X	9.5 GB
	X		X		9.6 GB
		X		X	13.1 GB
		X	X	X	12.8 GB
	X		X	X	16.6 GB
	X	X		X	19.7 GB
	X	X	X		14.7 GB
	X	X	X	X	20.6 GB

4.4 Symantec Clarity

Symantec Clarity is an automated, real-time IT service configuration management solution that automatically discovers application components in the infrastructure, dynamically maps their relationships, and tracks changes in real time. Use Operations Center's Symantec Clarity adapter to leverage this information and create a complete picture of the environment.

Use the following steps to create a Symantec Clarity adapter. Follow the links to sections that provide more detailed information.

To integrate Symantec Clarity:

1. Edit the Symantec `ClarityHierarchy.xml` file to customize the adapter hierarchy structure.

   For instructions, see Section 2.4, "Customizing the Adapter Hierarchy," on page 19.
2. Create a Symantec Clarity adapter in Operations Center.

   For instructions, see Section 2.1, "Creating an Adapter," on page 17.

   For property descriptions, see Table A-39, "Symantec Clarity Adapter Properties," on page 345.

4.5 Tideway Foundation

Tideway Foundation is an automated application dependency mapping tool used to index, model and map IT infrastructures. The integration with Operations Center enables automating the creation and maintenance of business service maps. When an incident occurs, use Foundation's up-to-date dependency maps to rapidly identify the root cause. Other business applications affected by the incident can be instantly identified, allowing fixes to be prioritized based on the criticality of the business process impacted.

While it is not recommended to run Tideway Foundation adapters in a clustered environment; if you do, it is necessary to create an adapter instance for each clustered server—when the same adapter instance is started and running from more than one clustered server, elements data is surfaced inconsistently.

Use the following steps to create a Tideway Foundation adapter. Follow the links to sections that provide more detailed information:

- Section 4.5.1, "Integrating Tideway Foundation," on page 158
- Section 4.5.2, "Creating an Appliance Definition," on page 159
- Section 4.5.3, "Using the Hierarchy File to Select Data," on page 160

4.5.1 Integrating Tideway Foundation

To create a Tideway Foundation adapter:

1. Place the `opencsv-1.7.jar` file in the `/OperationsCenter_install_path/classes/ext` directory.

   This file can be downloaded from: http://mirrors.ibiblio.org/pub/mirrors/maven/net.sf.opencsv/jars/opencsv-1.7.jar (http://mirrors.ibiblio.org/pub/mirrors/maven/net.sf.opencsv/jars/opencsv-1.7.jar)
2. Edit the `FoundationHierarchy.xml` file to customize the adapter hierarchy structure.

   For instructions, see Section 4.5.3, "Using the Hierarchy File to Select Data," on page 160.
3. Create a Tideway Foundation adapter in Operations Center.

   For instructions, see Section 2.1, "Creating an Adapter," on page 17.
4. Specify the Hierarchy File location and configure other adapter properties.

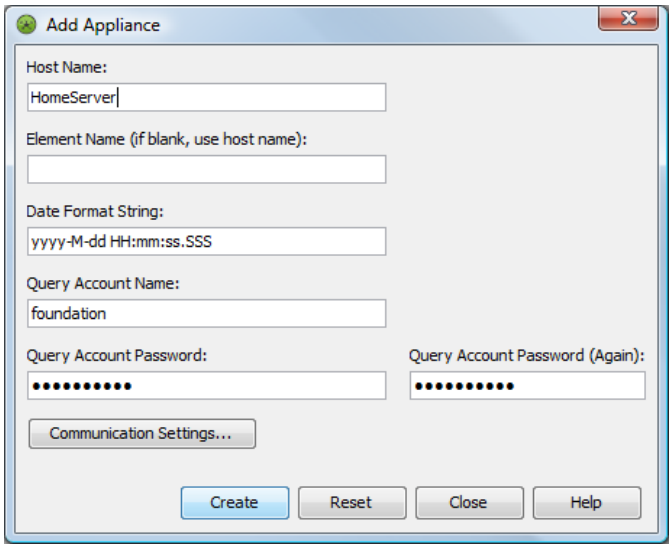
   For instructions, see Table A-40, "Tideway Foundation Adapter Properties," on page 346.
5. Continue to Section 4.5.2, "Creating an Appliance Definition," on page 159 and Section 4.5.3, "Using the Hierarchy File to Select Data," on page 160.

4.5.2 Creating an Appliance Definition

After starting the adapter, create definitions for appliances that are used to monitor traffic.

To create an appliance definition:

1. In the *Explorer* pane, expand *Elements* > the *Tideway Foundation* adapter element.
2. Right-click *Administration*, then click *Add Appliance* to open the Add Appliance dialog box:



3. Specify the following information to access the server hosting the appliance:

   Host Name: The IP address or host name where the appliance is installed.

   Element Name: A display name used for the appliance definition. Especially useful if an IP Address is the hostname.

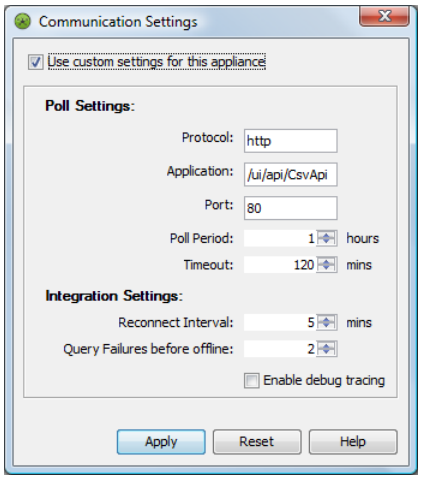
   Date Format String: The date format to use.

   Query Account Name: The Tideway account name.

   Query Account Password: The corresponding account password.

4. Click *Communication Settings* to define custom communication settings for the appliance.

   These settings monitor the downstream connection between the appliance and the client.



5 Select *Use custom settings for this appliance*.

6 Fill in the Poll Settings:

Protocol: The type of protocol used by the appliance.

Application: The application URL.

Port: The port number used by the appliance.

Poll Period: The duration of each polling performance. The recommended setting is 5 hours.

Timeout: The maximum number of minutes to attempt to perform a poll.

7 Fill in the Integration Settings:

Reconnect Interval: The number of seconds between attempts to reconnect to the appliance.

Query Failures before offline: The maximum number of query failures permitted before going offline.

8 Leave *Enable Debug Tracing* deselected.

If selected, a large amount of data is logged in the `formula.trc` file. Only select this option when asked by Support (http://www.netiq.com/support/) for troubleshooting.

9 Click *Apply* to save changes.

10 Click *Create* on the Add Appliance dialog box to add the new appliance.

The new appliance is added under the Elements, Tideway Foundation Adapter, Administration element. The adapter automatically explores the traffic across the switch to perform initial discovery.

4.5.3 Using the Hierarchy File to Select Data

The Tideway adapter hierarchy file (`FoundationHierarchy.xml`) can obtain the following information from Tideway and display it in Operations Center:

- Hosts
- Software Instances
- Business Application Instances
- Subnets
- Switches

If you are unfamiliar with the XML-based HierarchyFile structure used with various adapters, see Chapter 9, "Using the HierarchyFile," on page 227.

In addition, relationships between Software Instances, Business Applications Instances and Hosts can be displayed in Operations Center. The following relationships can be shown:

- Hosts belonging to a Business Applications Instances
- Software Instances belonging to a Business Applications Instances
- Software Instances belonging to a Host
- Hosts belonging to a Software Instance
- Business Application Instances belonging to Hosts
- Business Application Instances belonging to Software Instances

- "Understanding the <generator> Tag" on page 161
- "Displaying Host Properties" on page 163

Understanding the <generator> Tag

In the hierarchy file, the <generator> tag is used to dynamically create Operations Center elements from elements discovered in a different management system, such as Tideway Foundation. The field attribute compares specified fields in the incoming data with a target value, as a way to filter and select data.

Table 4-5 lists the valid field attributes and the corresponding Tideway nodes:

Table 4-5 *Corresponding XML Field Attributes and Tideway Nodes*

Field Attribute	Tideway Node
host.name	Host name
service.SPVI	Software Instance Name
service.BAI	Business Application Instance in the format:
	Business App. Name/Business App. Type/Product Version
service.Host	Host Name
BAI	Business Application Instance Name
SUBNET	Subnet Range
SWITCH	Switch

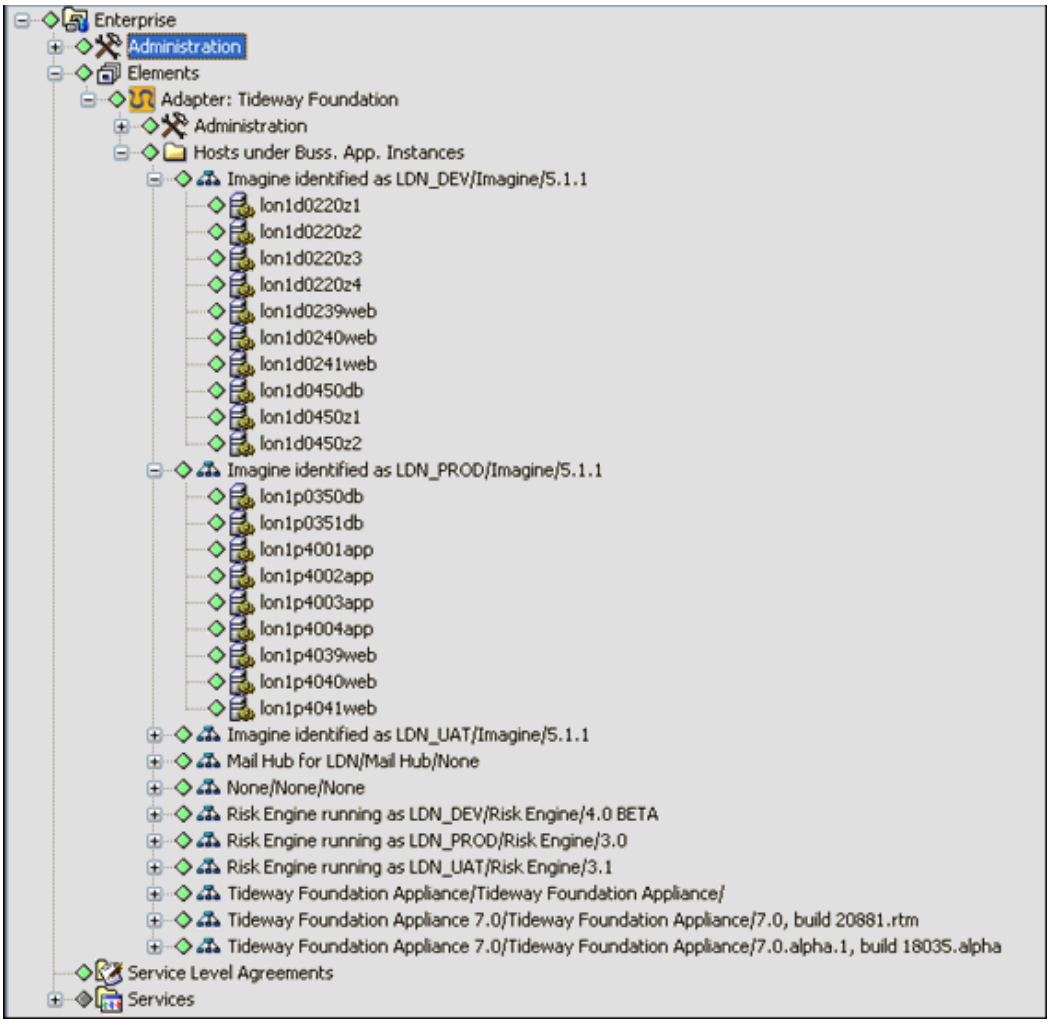
Relationships between nodes can be displayed by nesting <generator> tags in the Tideway hierarchy file. The <generator> tag field attributes that can create these relationships are the service types: service.SPVI, service.BAI. and service.Host.

For example, to show hosts that exist under a Business Application Instance, place the following entry in the Tideway hierarchy file:

```
<group class="gen_folder" name="Hosts under Buss. App. Instances">
  <generator class="formula_organizations" field="service.BAI" >
    <generator
class="admin_automation_server"
field="service.Host"
           />
  </generator>
</group>
```

The result in the Operations Center console is similar to Figure 4-4:

Figure 4-4 *Operations Center console Showing a Business Application Instance*

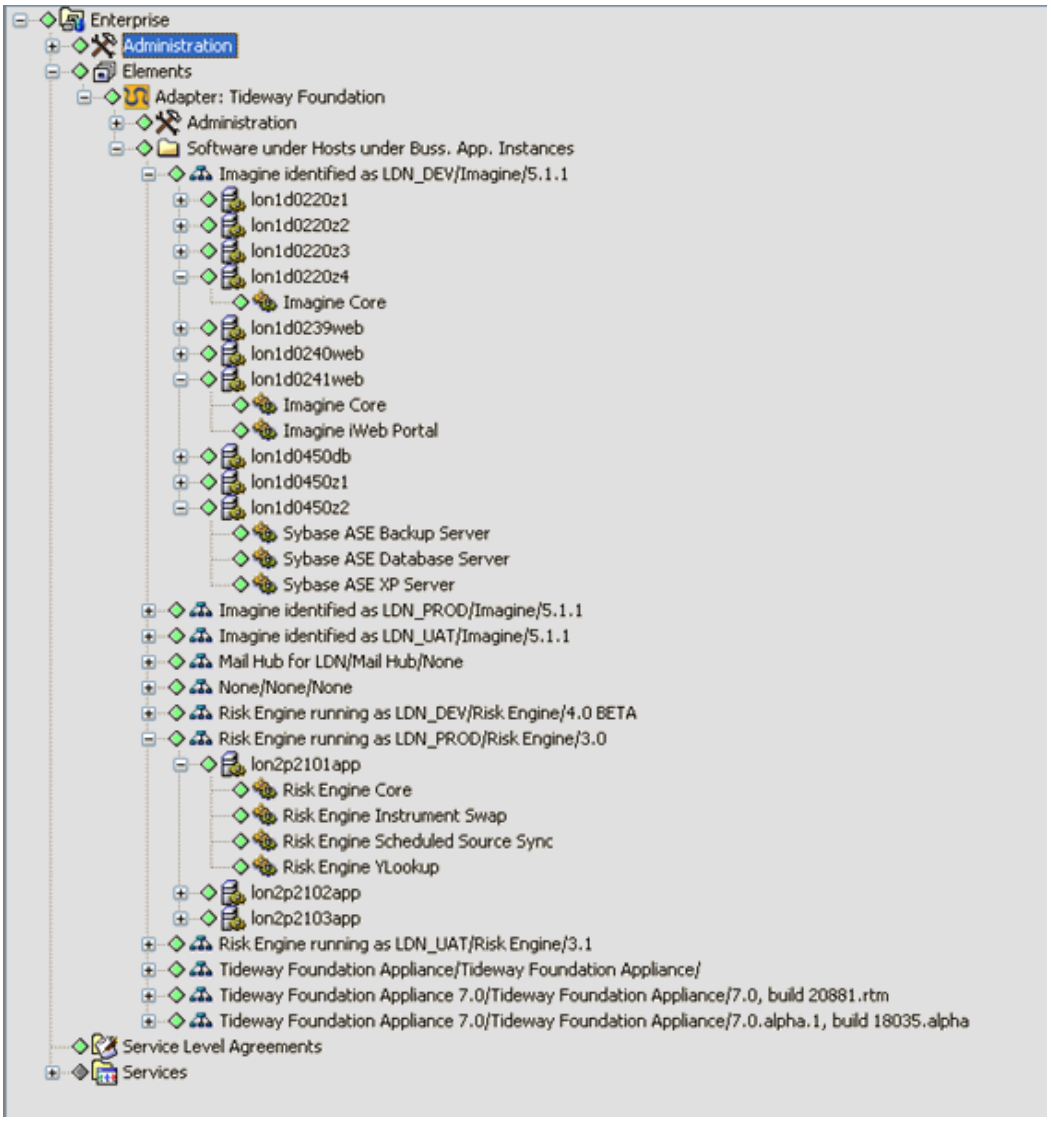


Another example shows the Software Instances that exist beneath the hosts (under Business Application Instances):

```
<group class="gen_folder" name="Software under Hosts under Bus. App. Instances">
  <generator class="formula_organizations" field="service.BAI" >
    <generator class="admin_automation_server" field="service.Host">
      <generator class="admin_automation" field="service.SPVI"/>
    </generator>
  </generator>
</group>
```

The resulting hierarchy is shown in Figure 4-5:

Figure 4-5 *Operations Center Console Showing Software Instances*



Nesting in any order is allowed for the service node. For example, service.BAI can be nested under service.Host.

Displaying Host Properties

The properties of a host can be viewed if both of the following apply:

- A hierarchy file with a `<generator>` tag contains a field attribute with a value of service.Host or host.name.
- The tag is not nested within another `<generator>` tag.

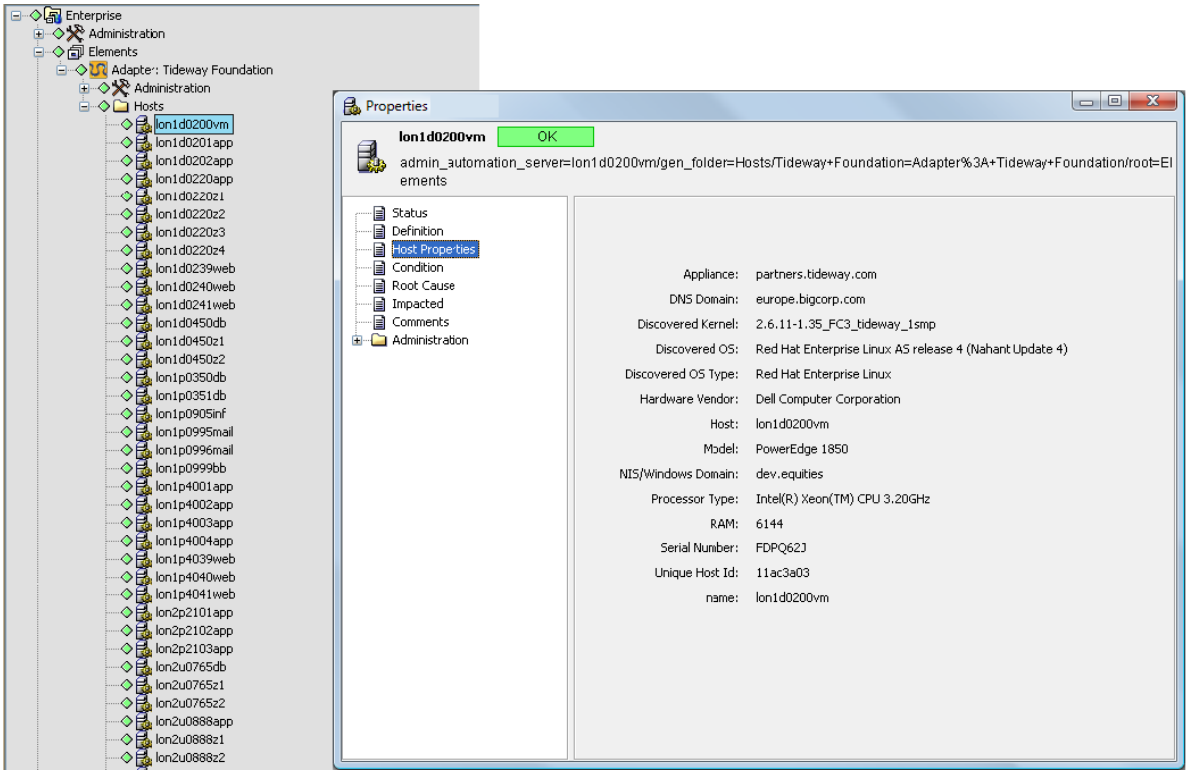
To view the Host Properties page, right-click a host node in the element hierarchy, click *Properties*, then click *Host Properties* in the left pane of the properties window.

For example, assume the following data is entered in the hierarchy file:

```
<group class="gen_folder" name="Hosts">
<generator class="admin_automation_server" field="service.Host" />
</group>
```

The resulting property page is shown in Figure 4-6:

Figure 4-6 *Property Page*



5 Trouble Ticket Systems Integrations

Tickets generated by trouble ticket systems display as alarms in Operations Center. Operations Center works with the trouble ticket systems listed in this section.

The following sections explain how to integrate with supported trouble ticketing systems:

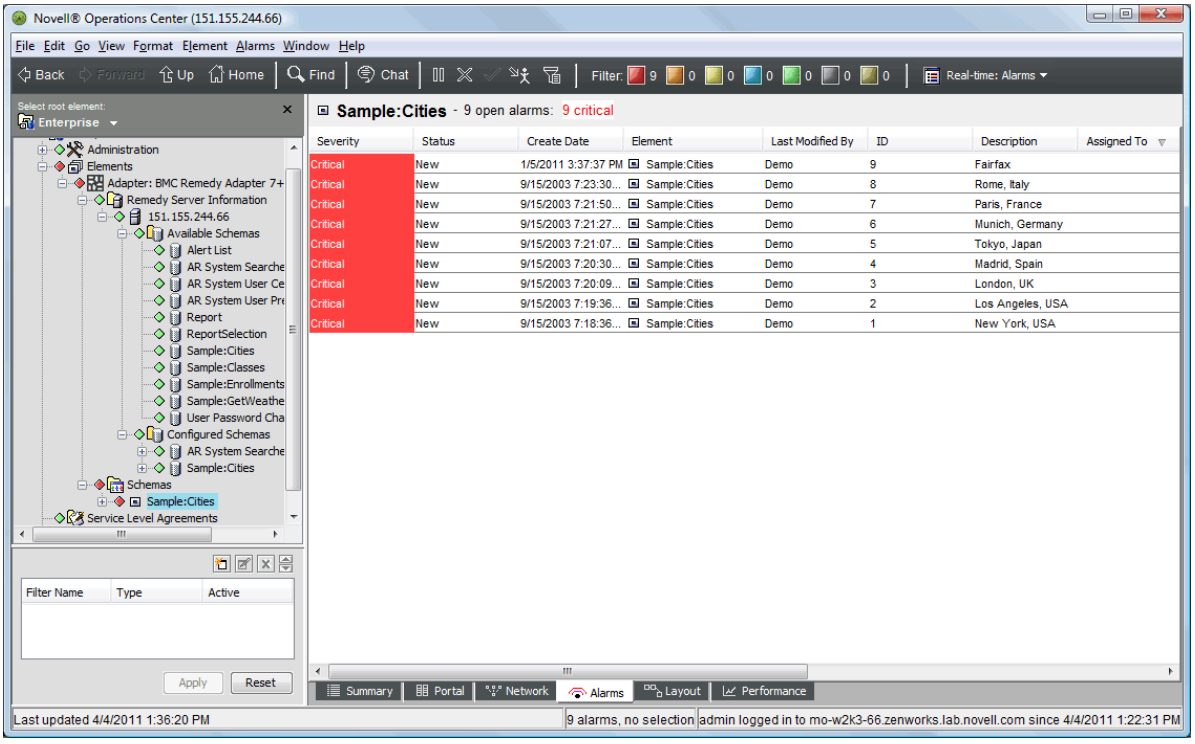
- Section 5.1, “BMC Remedy ARS Adapter,” on page 165
- Section 5.2, “HP ServiceCenter/Service Manager,” on page 177

For information about supported versions of these trouble ticket systems, see the *Operations Center 5.5 Getting Started Guide*.

5.1 BMC Remedy ARS Adapter

The BMC Remedy ARS adapter displays Remedy tickets as alarm information in Operations Center.

Figure 5-1 *Operations Center console: Remedy tickets are displayed as alarms*



In Operations Center, the BMC Remedy ARS adapter provides the following features:

- Maps alarm severity based on the value of a defined ticket field
- Filters tickets based on selected fields and values
- Easy configuration using industry-standard XML
- Groups tickets for quick and easy viewing via configurations in the hierarchy file
- Sets the polling frequency for updating ticket information

The follow sections describe configuration steps necessary to integrate with BMC Remedy ARS.

5.1.1 BMC Remedy ARS Requirements and Installation

The requirements to integrate BMC Remedy ARS:

- For a list of the supported versions, see Chapter 5, "Trouble Ticket Systems Integrations," on page 165.

  For more information about BMC Remedy ARS, see your *Remedy Action Request System Programmer's Guide* available from BMC.
- The AR System Java API, System C API library and `*.dll` files must be installed on the Operations Center server or the Operations Center remote container server.

  For more information, see "Setting up BMC Remedy 6.x and 7.0 on Windows" on page 166, "Setting up BMC Remedy 7.1 for Windows" on page 167, and "Setting Up a UNIX Environment" on page 168.
- Schemas must be Remedy Base Schemas. The Remedy Compound Schemas or Remedy Data Only Schemas cannot be used.

Setting up BMC Remedy 6.x and 7.0 on Windows

To set up BMC Remedy 6.x and 7.0 for Windows:

1. Copy the BMC Remedy JAR files to the `\OperationsCenter_install_path\classes\ext` directory:

   `\ARSystem_install\AR System\arapi70.jar`
   `\ARSystem_install\AR System\Arserver\Api\lib\arutil70.jar`
   `\ARSystem_install\AR System\Arserver\Api\lib\axis.jar`
2. DLL files from your Remedy installation must be copied to a directory on the Operations Center server or the Operations Center remote container server. For example, `\OperationsCenter_install_path\remedy_files\`. Be sure to verify this target directory is defined for the server's PATH environment variable.

   2a Copy the following DLL files from `\ARSystem_install\Arserver\Api\lib`:

   `arapi70.dll`

```
arjni70.dll
arutiljni70.dll
arxmlutil70.dll
```

2b Copy the following DLL files from *ARSystem_install*:

```
arrpc70.dll
arutl70.dll
icuuc20.dll
icudt20.dll
icuin20.dll
xerces-c_2_6.dll
```

2c Copy the following DLL files from *ARSystem_install*\AREmail:

```
icuuc32.dll
icudt32.dll
icuin32.dll
Xalan-C_1_9.dll
XalanMessages_1_9.dll
xerces-depdom_2_6.dll
```

3 Be sure to verify the target directory is defined for the server's PATH environment variable.

4 Download the `MSVCP71.dll` file from the Internet, or copy it from a BMC Atrium CMDB installation, to a directory on the Operations Center server.

Be sure to verify the target directory is defined for the server's PATH environment variable.

5 Restart the Operations Center server.

For instructions, see "Manually Starting the Operations Center Server" and "Starting the Operations Center Server in UNIX" in the *Operations Center 5.5 Server Installation Guide*.

Setting up BMC Remedy 7.1 for Windows

To set up BMC Remedy 7.1 for Windows:

1 Copy the following files from *ARSystem_install*\AR System\Arserver\ to *OperationsCenter_install_path*\integrations\ext\BMCRemedy:

```
Api\lib\arapi71.jar
Api\lib\arcmn71.jar
Api\lib\arrpc71.jar
Api\lib\commons-codec-1.3.jar
Api\lib\commons-collections-3.2.jar
Api\lib\commons-configuration-1.3.jar
Api\lib\commons-digester-1.7.jar
Api\lib\commons-lang-2.2.jar
Api\lib\oncrpc.jar
Api\lib\spring.jar
xercesImpl.jar
```

2 DLL files from your Remedy installation must be copied to a directory on the Operations Center server or the Operations Center remote container server. For example, `\OperationsCenter_install_path\remedy_files\`. Be sure to verify this target directory is defined for the server's PATH environment variable.

   2a Copy the following DLL files from `\ARSystem_install\server_name\Arserver\Api\lib`:

   ```
   arapi71.dll
   arjni71.dll
   arrpc71.dll
   ```

   2b Copy the following DLL files from `\ARSystem_install\server_name`:

   ```
   icuinbmc32.dll
   icuucbmc32.dll
   icudt32.dll
   arutl71.dll
   ```

3 Download the `MSVCP71.dll` file from the Internet, or copy it from a BMC Atrium CMDB installation, to a directory on the Operations Center server. Be sure to verify the target directory is defined for the server's PATH environment variable.

4 Restart the Operations Center server.

   For instructions, see "Manually Starting the Operations Center Server" and "Starting the Operations Center Server in UNIX" in the *Operations Center 5.5 Server Installation Guide*.

Setting up BMC Remedy 8.1 for Windows

To set up BMC Remedy 8.1 for Windows:

1 Copy the following files from `\ARSystem_install\AR System\Arserver\` to `\OperationsCenter_install_path\integrations\ext\BMCRemedy8`:

   ```
   arapi81_build001.jar
   log4j-1.2.14.jar
   ```

2 Restart the Operations Center server.

   For instructions, see "Manually Starting the Operations Center Server" and "Starting the Operations Center Server in UNIX" in the *Operations Center 5.5 Server Installation Guide*.

Setting Up a UNIX Environment

To set up on Solaris, and RedHat Linux:

1 Copy the BMC Remedy JAR files to the Operations Center server.

- For BMC Remedy versions 6.x, and 7.0, copy the following files to `/OperationsCenter_install_path/classes/ext`:

```
/ARSystem_install/AR System/arapi70.jar

/ARSystem_install/AR System/Arserver/Api/lib/arutil70.jar

/ARSystem_install/AR System/Arserver/Api/lib/axis.jar
```

- For BMC Remedy version 7.1, copy the following files from `/ARSystem_install/AR System/Arserver/` to `/OperationsCenter_install_path/integrations/ext/BMCRemedy`:

```
Api/lib/arapi71.jar
Api/lib/arcmn71.jar
Api/lib/arrpc71.jar
Api/lib/commons-codec-1.3.jar
Api/lib/commons-collections-3.2.jar
Api/lib/commons-configuration-1.3.jar
Api/lib/commons-digester-1.7.jar
Api/lib/commons-lang-2.2.jar
Api/lib/oncrpc.jar
Api/lib/spring.jar
xercesImpl.jar
```

- For BMC Remedy version 8.1, copy the following files from */ARSystem_install/*`ARSystem/Arserver/` to */OperationsCenter_install_path/*`integrations/ext/BMCRemedy8`:

```
arapi81_build001.jar
log4j-1.2.14.jar
```

2. For BMC Remedy versions 6.x, and 7.0 on Solaris and Linux, copy all `.so` and `.32*` files into the `LD_LIBRARY_PATH` directory.
3. Copy the C API libraries to their defined environment variable for library path:

HP-UX:	Copy all `.sl` files into the `SHLIB_PATH` directory.
AIX:	Copy all `.so` files into the `LIBPATH` directory.

4. Change the permissions of each `.so` or `.sl` file by performing a `chmod 755 filename` command.
5. Restart the Operations Center server.

   For instructions, see "Manually Starting the Operations Center Server" and "Starting the Operations Center Server in UNIX" in the *Operations Center 5.5 Server Installation Guide*.

5.1.2 Creating a Remedy Adapter

The following provides the basic steps for creating a BMC Remedy ARS adapter with links to sections that provide detailed information.

To create a BMC Remedy ARS adapter:

1. Edit the adapter hierarchy XML file to customize the adapter hierarchy structure.
   - For BMC Remedy ARS v6.x and v7.0, edit `RemedyHierarchy.xml`.
   - For BMC Remedy ARS v7.1, 7.2 or 8.1, edit `BMCRemedyHierarchy.xml`.

   For instructions, see Section 5.1.3, "Customizing the Adapter Hierarchy," on page 170.
2. Create the BMC Remedy ARS adapter in Operations Center.
   - For BMC Remedy ARS v 6.x and v7.0, select *Remedy Action Request System* for the adapter type.
   - For BMC Remedy ARS v7.1 or 7.2, select *BMC Remedy Adapter 7+* for the adapter type.
   - For BMC Remedy ARS v8.1, select *BMC Remedy Adapter 8* for the adapter type.

   For instructions, see Section 2.1, "Creating an Adapter," on page 17.

3 Edit the adapter configuration file to specify schemas, field and ticket filters, and alarm field mappings.

   For instructions, see Section 5.1.4, "Understanding the Remedy Configuration File," on page 171 and Section 5.1.5, "Updating the Remedy Configuration," on page 174.

   Note that after updating the configuration file by modifying the configuration file directly, or by using *Edit Remedy Configuration* or *Show Server Information* right-click options, the adapter must be restarted.

4 Edit the Remedy Action Request System adapter properties.

   For property descriptions, see Section A.39, "Symantec Clarity," on page 345.

   Because each BMC Remedy ARS implementation is different, configure the adapter to reflect the specific implementation. For example, use the `AlarmColumns` adapter property to contain additional noncore fields of the associated schema.

   If the Port-Mapper is not running on the Remedy server, and the system uses version 7.*x* or later of the Remedy API JAR files, you must configure the Server Port adapter property to the port number used by the Remedy ARS server. Otherwise, the integration fails.

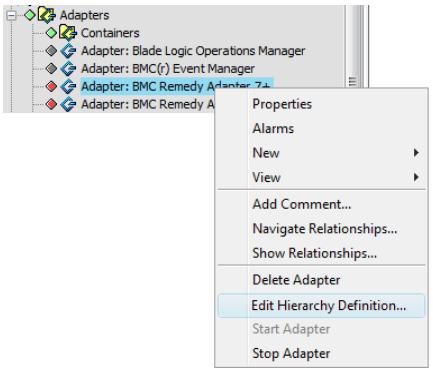
5.1.3 Customizing the Adapter Hierarchy

Modify the adapter hierarchy file using the Operations Center XML editor to customize the adapter.

- For BMC Remedy ARS versions 6.x and 7.0, edit `RemedyHierarchy.xml`.
- For BMC Remedy ARS version 7.1, edit `BMCRemedyHierarchy.xml`.

For example, set up the adapter hierarchy to group tickets by schemas. Then within a schema, group tickets by assignment and status.

Figure 5-2 *Edit the Hierarchy Definition from the Adapter Right-Click Menu*



For more information about the Operations Center XML editor, see *The Operations Center XML Editor* in the *Operations Center 5.5 Server Configuration Guide*.

For more information about hierarchy files, see Chapter 9, "Using the HierarchyFile," on page 227.

5.1.4 Understanding the Remedy Configuration File

The adapter configuration file enables full customization of the BMC Remedy ARS adapter to surface ticket information in Operations Center to specify:

- Schemas to retrieve
- Additional noncore Remedy fields to retrieve for each schema
- Tickets to retrieve for each schema (i.e. retrieve only open tickets)
- Field to map to the severity of the Operations Center alarm for each schema

  By default, all alarms have a severity of `informational`.
- Alarm filtering based on field ID and value

Edit the following configuration files:

- For BMC Remedy ARS versions 6.x and 7.0, edit `RemedyConfiguration.xml`.
- For BMC Remedy ARS version 7.1, 7.2, and 8.1 edit `BMCRemedyConfiguration.xml`.

The RemedyConfiguration_1.0.dtd data dictionary describes the exact syntax of the configuration file.

The current integration with Remedy does not support Diary fields. Errors contain the message "Cannot specify a diary field."

Defining Schemas

The BMC Remedy ARS adapter works with schemas of type Remedy Base Schemas only.

Base schemas have a set of eight core fields and zero or more additional fields. Implementations commonly use this schema type to maintain Remedy tickets. Other unsupported types are Compound Schemas and Data Only Schemas.

For more details about Remedy Schema types, see the Remedy documentation.

In the configuration file, define all schemas that Operations Center retrieves. The minimum definition is:

```
<schema name="AR 4.0 Sampler" />
```

However, this definition does not filter or map severities and only extracts core fields.

Specifying Additional Fields

BMC Remedy ARS schemas based on Remedy Base Schemas contain eight core fields:

- *Entry ID (ticket ID)*
- *Assigned to*
- *Date created*
- *Date last modified*
- *Last modified by*
- *Description*
- *Status*
- *Submitted by*
- *Short Description*

Optionally specify one or more additional fields in the `RemedyConfiguration.xml` or `BMCRemedyConfiguration.xml` file. These fields depend on the specific implementation of the schema.

```
<schema name="AR 4.0 Sampler">
  <field id="536870926"/>
  <field id="536870931"/>
</schema>
```

Using the field element, the `id=` attribute represents the field ID from the Remedy FIELDS table. For example, add a field with ID 536870926 to also view the Box 10 field of the ticket.

Filtering Tickets

By default, Operations Center retrieves all tickets of a schema. The BMC Remedy ARS adapter allows specifying a simple filter that determines a subset of the tickets to retrieve. For example, set up the adapter to retrieve and maintain open tickets only. The following filter does not display any open tickets:

```
<schema name="AR 4.0 Sampler">
  <filter field_id="7" operator="less" value="4"/>
</schema>
```

The *_id* field is for status and only values less than 4 display.

Querying Tickets

The `RemedyConfiguration.xml` and `BMCRemedyConfiguration.xml` files allows query qualification (filtering) using the `<query>` XML tag.

In the following example, the `<filter>` filtering mechanism restricts the Remedy records returned to those records whose field 7 value is less than 15. An additional restriction uses the `<query>` XML tag to select only those records containing a *Status* field equal to 2 and an *Asset Name* field equal to `TEST`.

```
<schema name="CHG:Change" enable="true">
      <field id="240000011" polled="true"/>
      <field id="200000020" polled="true"/>
      <filter field_id="7" operator="less" value="15"/>
      <query value="&apos;Status&apos; = 2 AND &apos;Asset Name&apos; =
&quot;TEST&quot;">
          <field name="Asset Name"/>
          <field id="7"/>
      </query>
      <mapping field_id="7" default="UNKNOWN">
         <sev_map value="0" severity="CRITICAL"/>
         <sev_map value="1" severity="INFO"/>
         <sev_map value="2" severity="MINOR"/>
         <sev_map value="3" severity="MAJOR"/>
         <sev_map value="4" severity="OK"/>
      </mapping>
   </schema>
```

The `<query>` XML tag must have sub `<field>` tags for the fields that are referenced in the query text value. In the previous example, the sub `<field>` tags are:

```
<field name="Asset Name"/>
<field id="7"/>
```

Use the `name=` attribute if the field was already referenced by `id=` in the `<schema>` level `<field>` XML tag. In the example, it is:

```
<field id="240000011" polled="true"/>
```

Use the `id=` attribute if the field was not referenced in a `<schema>` level `<field>` XML tag.

Mapping Alarm Severity

Core BMC Remedy ARS fields do not contain a severity field and do not allow Operations Center to derive a severity from them.

However, most custom implementations contain a noncore field that indicates the urgency of a ticket. Use this field to map to an alarm severity in Operations Center.

For example, a schema might contain the urgency field, which can have a value of low, medium or high. Then set up the `RemedyConfiguration.xml` or `BMCRemedyConfiguration.xml` file to display in Operations Center all tickets with low urgency as MINOR alarms, medium urgency as MAJOR alarms and the high urgency as CRITICAL alarms.

```
<schema name="AR 4.0 Sampler">
  <mapping field_id="7" default="MAJOR">
    <sev_map value="0" severity="CRITICAL"/>
    <sev_map value="1" severity="INFO"/>
    <sev_map value="2" severity="MINOR"/>
    <sev_map value="3" severity="MAJOR"/>
    <sev_map value="4" severity="OK"/>
  </mapping>
</schema>
```

The mapping above uses the STATUS field_id 7 for mapping severities.

Mapping Alarm Date/Time Values

By default, the alarm Date/Time value displayed in Operations Center is mapped to the Remedy core field named *Modified Date*. It is possible to map the Date/Time to a different Remedy core field. For example, map Date/Time to the *Create Date core* field in Remedy, assuming Create Date has a field ID of 3. Add this line to the schema definition:

```
<last_update field_id="3" />
```

5.1.5 Updating the Remedy Configuration

Update the Remedy Configuration file using Operations Center menu options. Also edit the Remedy Configuration file using the XML Editor.

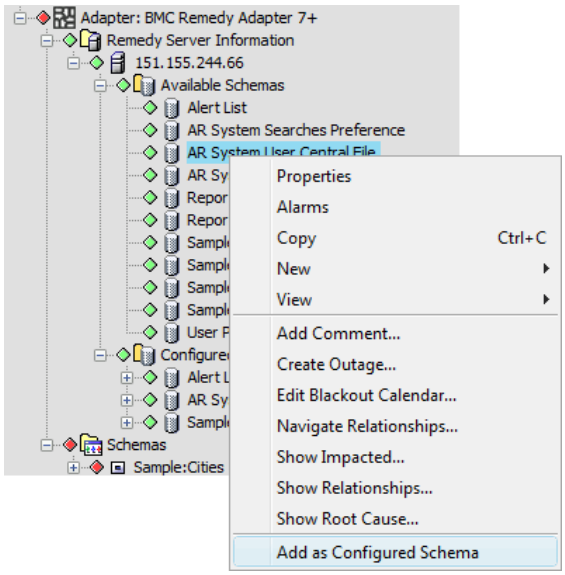
Note that after updating the configuration file by modifying the configuration file directly, or by using *Edit Remedy Configuration* or *Show Server Information* right-click options, the adapter must be restarted.

- "Adding Schema and Fields Through the Remedy Adapter" on page 174
- "Adding a Field" on page 175
- "Adding All Fields Under a Selected Category" on page 175
- "Removing a Schema Through the Remedy Adapter" on page 175
- "Editing the Remedy Configuration File Using the XML Editor" on page 176

Adding Schema and Fields Through the Remedy Adapter

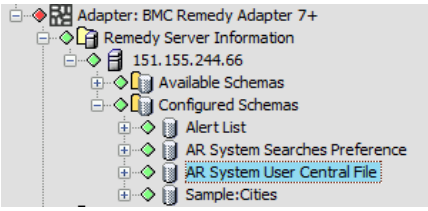
To add schema and fields through the Remedy adapter:

1 In the *Explorer* pane, expand *Elements* > the Remedy adapter > *Remedy Server Information* > the server name > *Available Schemas*.



2 Right-click a schema, then click *Add as Configured Schema*.

The selected schema displays under Configured Schemas in the *Explorer* pane:



Core fields automatically display under *Configured Fields* > *Core Fields* in the *Explorer* pane.

Adding a Field

To add a field:

1 In the *Explorer* pane, right-click a field under *Available Fields*, then click *Add as Configured Field*.

The selected field displays under *Configured Fields* > *Additional Fields*.

Adding All Fields Under a Selected Category

To add all fields under a selected category, such as *By Data Type*:

1 Right-click the parent element, then click *Auto Add Fields*.

All associated fields display under *Configured Fields* > *Additional Fields*.

Removing a Schema Through the Remedy Adapter

To remove a configured schema through the Remedy adapter:

1 In the *Explorer* pane, expand *Elements* > *Remedy Adapter* > *Remedy Server Information* > the server name > *Configured Schemas*.

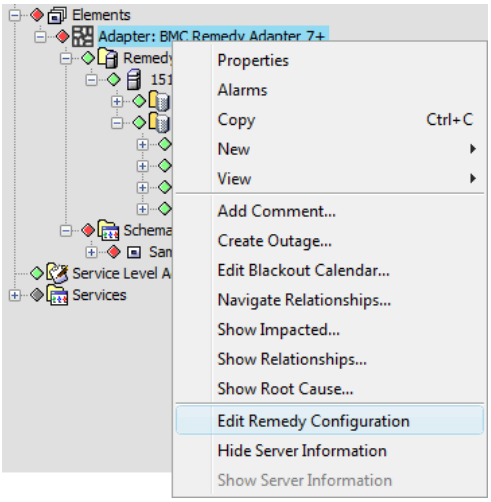
2 Right-click a schema, then click *Remove as Configured Schema*.

The schema is removed and no longer applies to the adapter. Alarms are removed for the removed schema. To remove the elements, the adapter must be restarted.

Editing the Remedy Configuration File Using the XML Editor

To edit the Remedy configuration file using XML adapter:

1. In the *Explorer* pane, expand *Elements*.
2. Right-click *Remedy Adapter*, then click *Edit Remedy Configuration* to open the XML Editor dialog box.



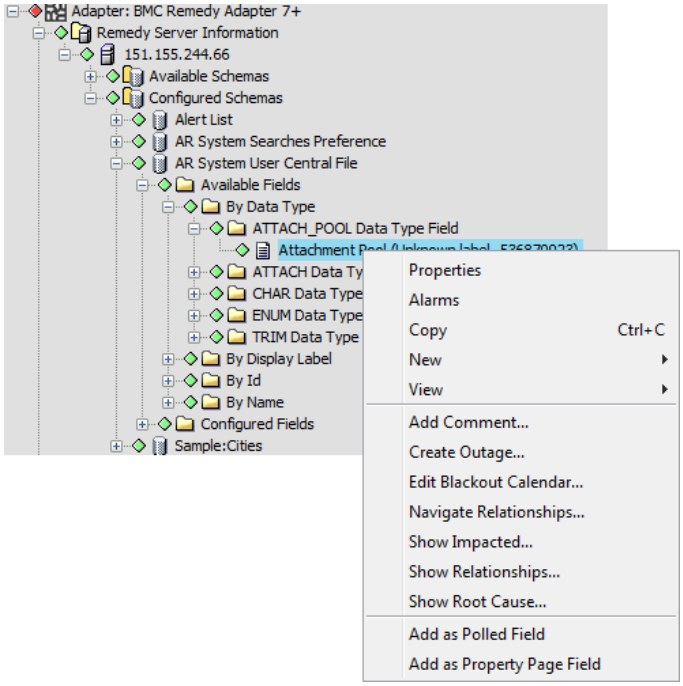
3. Edit and save the configuration file.
4. Restart the adapter.

5.1.6 Configuring Schema Fields

To configure the fields for a schema that is retrieved in the Operations Center, specify the field type by performing these steps:

1 In the *Explorer* pane, expand the *Elements* root element > *Remedy Adapter* > *Remedy Server Information* > a server name > *Configured Schemas* > a schema name > *Available Fields*.

2 Right-click a field, then click *Add as Polled Field* or *Add as Property Page Field*:



Polled fields can be displayed as alarm columns by adding them to the *Alarm Columns* adapter property. Property page fields are retrieved when the alarm *Additional Fields* property page displays.

For more information about BMC Remedy adapter properties, see Section A.3, “BMC Remedy Action Request System (ARS),” on page 282.

3 Configure a minimum number of fields as polled fields.

The existence of a large number of polled fields can slow down the polling process. Configure fields as property page fields if they need not display in an alarm column or be collected as part of BSA, and so on.

5.2 HP ServiceCenter/Service Manager

HP ServiceCenter® and HP Service Manager® do not require ORB software to integrate. However, it is necessary to create an adapter for each instance of ServiceCenter and Service Manager on the network.

The HP ServiceCenter and HP Service Manager adapters have the exact same functionality as the previously named Peregrine Systems ServiceCenter adapter.

5.2.1 Integrating ServiceCenter and Service Manager

To integrate to ServiceCenter or Service Manager:

1. Perform the required customizations to the ServiceCenter server and restart the SOAP Server.

   For instructions, see Section 5.2.2, “Configurations for ServiceCenter and Service Manager,” on page 178.

2. Create an adapter for each instance of a ServiceCenter or Service Manager on the network.

   For instructions, see Section 2.1, “Creating an Adapter,” on page 17. For property descriptions, see Table A-18, “HP ServiceCenter and HP Service Manager Adapter Properties,” on page 309.

   Have available the following information:

   - The host name of the ServiceCenter or Service Manager instance
   - The port that the ServiceCenter or Service Manager listens on (default is 12700)
   - The user name and password for a valid user account on ServiceCenter or Service Manager

   The HP ServiceCenter v6.2 requires the use of the `/OperationsCenter_install_path/database/examples/ServiceCenterRel62Configuration.xml` file.

   The HP Service Manager requires the use of the `/OperationsCenter_install_path/database/examples/ServiceManagerConfiguration.xml` file.

   The HP Service Manager 9.3 requires the use of the `/OperationsCenter_install_path/database/examples/ServiceManagerConfiguration_9.3.xml` file.

3. Customize the integration to surface new modules and define alarm operations in the adapter’s configuration XML file.

   For information about defining or customizing modules in the configuration XML file, see Section 5.2.3, “Defining Modules and Alarm Operations,” on page 182.

5.2.2 Configurations for ServiceCenter and Service Manager

Operations Center ServiceCenter adapters use the ServiceCenter/Service Manager’s Web Services interface to send requests to and receive data from the ServiceCenter/Service Manager server. The adapter uses a polling technique to refresh alarm data.

Configuring the ServiceCenter WSDL

For efficiency, the ServiceCenter adapter uses a time stamp to return alarm data for items that changed since the previous poll. Configure the ServiceCenter WSDL definition for the Web Services interface to include this time stamp field.

To configure the WSDL to allow external access to Operations Center required fields:

1. Using the ServiceCenter client, open the *System Navigation* dialog box.
2. Click *Menu Navigation* > *Toolkit* to expand them.
3. Double-click *WSDL Configuration* to open the *External Access Definition* dialog box.
4. Click *Search*.

   A list of ServiceCenter tables configured for external access displays.

   The adapter must have access to the following tables:

ServiceCenter Table	Operations Center Module	ServiceCenter Web Service
cm3r	Change	ChangeManagement.wsdl
device	Inventory	ConfigurationManagement.wsdl
incidents	Service	ServiceDesk.wsdl
probsummary	Incident	IncidentManagement.wsdl

5. Click the *Data Policy* tab.
6. In the cm3r, device, incidents, and probsummary tables, locate the *sysmodtime* field and perform the following changes:
   - Set the *API Caption* column value to sysmodtime.
   - Set the *Exclude* column value to False.
   - Set the *API Data Type* column value to DateTimeType.
   - Save the changes for all tables.
7. In the incidents table, delete the StringType value that is entered for the *API Data Type* column for the following fields and then save the changes:
   - *Description*
   - *Resolution*
   - *Update.action*
8. In the *probsummary* table, locate the *Status* field and perform the following changes:
   - **8a** Set the *API Caption* column value to `Status`.
   - **8b** Set the *Exclude* column value to False.
   - **8c** Save the changes.
9. Click the *Allowed Actions* tab.
10. In the incidents table where *Allowed Actions* is equal to `Clone`, set the *Action Names* column to `Clone`.

    This ensures the `ServiceDesk.wsdl` generates correctly.
11. Restart the ServiceCenter server.

For instructions, see "Manually Starting the Operations Center Server" and "Starting the Operations Center Server in UNIX" in the *Operations Center 5.5 Server Installation Guide*.

12 Configure the adapter (in the `ServiceCenter Port` property) to send its SOAP requests to the port specified in the `system:port_number` command in ServiceCenter's `sc.ini` file.

For information about troubleshooting SOAP traffic, see "Debugging SOAP Messages" on page 181.

For more information about the `ServiceCenter Port` property, see the Table A-18, "HP ServiceCenter and HP Service Manager Adapter Properties," on page 309.

Configuring the Service Manager WSDL

For any Service Manager fields referenced in the adapter's configuration XML file, Service Manager must be configured to expose those fields in ServiceManager through its External Access Definition WSDL Configuration feature.

To configure the WSDL to allow external access to Operations Center required fields:

1 Using the Service Manager client, open the *System Navigation* dialog box.

2 Click *Menu Navigation* > *Tailoring* to expand them.

3 Double-click *WSDL Configuration* to open the *External Access Definition* dialog box.

4 Click *Search*.

A list of Service Manager tables configured for external access displays.

For the example configuration XML files that ships with Operations Center, the adapter must have access to the following tables

Service Manager Table	Operations Center Module	ServiceCenter Web Service
Change/cm3r	Change	ChangeManagement.wsdl
Device/device	Configuration	ConfigurationManagement.wsdl
Incident/ probsummary	Incident	IncidentManagement.wsdl
Interaction/incidents	Service	ServiceDesk.wsdl

5 Click the desired Service Manager table.

6 Click the *Fields* tab.

7 In the adapter's `configuration.xml` file, verify each Service Manager field referenced between the `<field>` and `</field>` xml for each module is listed in the appropriate Service Manager table.

Examples of the configuration files are located in the `/OperationsCenter_install_path/database/examples` directory. Make a copy of the appropriate xml file before customizing. The customized configuration file must be specified in the adapter's `Configuration File` property. See Table A-18, "HP ServiceCenter and HP Service Manager Adapter Properties," on page 309.

8 If a field is missing, do the following:

8a Click the blank entry at the bottom of the table in the *Field* column.

8b Select the field to add from the drop-down list.

8c Click in the *Caption* field.

8d Specify the caption. This must match the case-sensitive field name specified in the adapter's configuration XML file.

8e For the *sysmodtype* field, click in the *Type* column, and specify `DateTimeType`.

9 Repeat Step 5 to Step 8 for each table needed.

10 Configure the adapter (in the `Service Manager Port` property) to send its SOAP requests to the port specified in the `system:`*`port_number`* command in Service Manager's `sm.ini` file.

For information about troubleshooting SOAP traffic, see "Debugging SOAP Messages" on page 181.

For more information about the he Service Manager Port adapter property, see the Table A-18, "HP ServiceCenter and HP Service Manager Adapter Properties," on page 309.

Debugging SOAP Messages

To debug SOAP messages, optionally start another instance of the ServiceCenter server. This is described in the "Debugging SOAP messages" or "SOAP Messages" section of the ServiceCenter and Service Manager help and partially printed below for reference.

To debug SOAP messages:

1 At the command line, enter one of the following:

- For ServiceCenter:

  `scenter -apiserver:`*`unique_port_number`* `-debughttp -log:../logs/debug.log`

  where `-apiserver:`*`unique_port_number`* identifies a port where only this process runs, and `-log:../logs/debug.log` defines a path to store the logs (`/logs/debug.log` are examples).

  Normally, all ServiceCenter processes for a particular installation use parameter values from `sc.ini` in the ServiceCenter `RUN` directory, and all share the `sc.log` file that is specified in `sc.ini`. Manually starting a new ServiceCenter instance with a different log parameter value in the command line causes one ServiceCenter process to run in isolation and produce separate debug output.

  Select a port number that is not likely to be used by other running process.

- For Service Manager:

  `sm -apiserver:`*`unique_port_number`* `-debughttp:1 -log:../logs/debug.log`

  where `-apiserver:`*`unique_port_number`* identifies a port where only this process runs, and `-log:../logs/debug.log` defines a path to store the logs (`/logs/debug.log` are examples).

  Normally, all Service Manager processes for a particular installation use parameter values from `sm.ini` in the Service Manager `RUN` directory, and all share the `sc.log` file that is specified in `sm.ini`. Manually starting a new Service Manager instance with a different log parameter value in the command line causes one Service Manager process to run in isolation and produce separate debug output.

  Select a port number that is not likely to be used by other running process.

2 Recreate the problem you are trying to debug.

3 Search the `RECV.LOG` file for the incoming message.

4 Search the `SEND.LOG` file for the outgoing response.

For example, to start a ServiceCenter server on port 12700 and record the log output in the `/logs/scsoap.log` file, enter the following command:

```
scenter -apiserver:12700 -log:../logs/scsoap.log
```

This example configures the adapter to send its SOAP requests to port 12700.

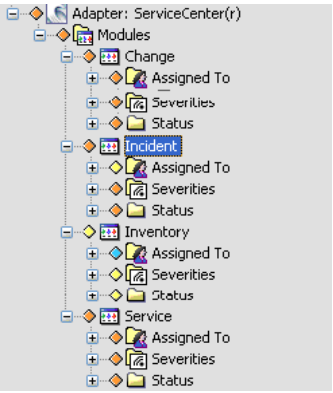
If the SOAP server is not running or loses its connection, in Operations Center the condition indicator for the adapter changes to CRITICAL (red) and the following message displays:

```
Could not connect to ServiceCenter while testing the connection by loading URL:
http://servername:12700/IncidentManagement.wsdl: Connection refused.
```

5.2.3 Defining Modules and Alarm Operations

The flexibility of HP's ServiceCenter/Service Manager API allows the modules and alarm operations surfaced by the adapter to be customized as required using the adapter's configuration and hierarchy XML files.

Figure 5-3 *ServiceCenter Adapter Default Hierarchy*



Operations Center uses a `configuration.xml` file to specify how the adapter interacts with the ServiceCenter/Service Manager Soap server to return information about requested modules. These files define:

- The modules represented as elements in the adapter hierarchy
- An alarm severity mapping
- A mapping of standard alarm fields
- Any custom alarm or element operations

By default, the `configuration.xml` define access for the *Change*, *Incident*, *Inventory*, and *Service* modules.

Whereas, the `hierarchy.xml` file specifies the hierarchical structure for each module. By default, module information is grouped by *Assigned to*, *Severities*, and *Status* values.

Customizing the `configuration.xml` file requires an understanding of SOAP requests and XML, as well as ServiceCenter/Service Manager security and APIs.

The following sections provide details on defining and customizing module declarations to surface elements and alarms:

Specifying Modules to Define Elements and Retrieve Alarms

The adapter uses ServiceCenter/Service manager modules, defined with module tags, to retrieve records and convert them into alarms in Operations Center. Each defined module displays as an element under the adapter.

To define a new module:

1. In a text editor, open the */OperationsCenter_install_path*`/database/examples/ServiceCenterConfiguration.xml` or `ServiceManagerConfiguration.xml` file (or `ServiceManagerConfiguration_9.3.xml` for Service Manager 9.3).
2. Define the new module and setup properties.

   ```
   <module name="ModuleName" enable="true">
     <properties>
     </properties>
   </module>
   ```

   It is recommended to start with an existing and working example (copy and paste the *Configuration* module example provided in the default configuration.xml file) and modify when creating a new module.

   For information about the Module XML Structure, see "Understanding the Module XML Structure" on page 184.

   For information about the XML tags used, see "Module XML Tags Reference" on page 192.
3. Inside a `formula` tag, use `fields` and `severities` tags to define how ServiceCenter/Service Manager fields are to be mapped to Operations Center alarms and define severities.

   For information on defining alarm fields and severities, see "Mapping Alarm Fields and Severities" on page 184
4. Inside a `soap` tag, configure the SOAP request.

   For information on defining the connection between ServiceCenter/Service Manager and Operations Center, see "Configuring the SOAP Request" on page 186.
5. Inside an `operations` tag, configure both required and optional operations.

   `pollKeys` and `getPollRecords` tags are used to define required operations necessary to return a specific set of data. Other operations are often used to define right-click operations on alarms.

For more information about configuring operations, see "Setting Up Required Operations" on page 187 and "Defining Custom Alarm Operations" on page 191.

6 Save the file.

7 Restart any adapters that use the configuration XML file.

Understanding the Module XML Structure

The following is the XML structure used by each module definition:

```
<module name="moduleName" enable="true">
    <properties>
     <formula>
          <fields></fields>
          <severities></severities>
     </formula>
     <soap>
       <endpoint></endpoint>
         <namespace></namespace>
       <server></server>
       <port></port>
         <username></username>
         <password></password>
         <operations></operations>
     </soap>
    </properties>
</module>
```

Tags inside the `formula` tag are used to define the ServiceCenter/Service Manager fields used to populate Operations Center alarms as well as set their severity. Tags inside the `soap` tag are used to setup a SOAP connection between the ServiceCenter/Service Manager server and Operations Center as well as define required and optional operations.

For more information about module XML tags, see "Module XML Tags Reference" on page 192.

Mapping Alarm Fields and Severities

Setting subelements for the `formula` tag, in the configuration.xml file, allows you to map ServiceCenter/Service Manager fields to adapter alarms and specify severities.

Defining Alarm Fields

The `fields` tag predefines the ServiceCenter/Service Manager fields that are used for adapter alarm fields. The following 5 field definitions are required for normalization:

- **status:** The field used by the severities mapping tag.

- **lastUpdate:** The field used for the alarm time stamp when last updated. Allows Managed Object to query for newly updated records only.
- **key:** The field used for the alarm identifier.
- **description:** The field used for the description of the alarm.
- **assignedTo:** The field indicating person the alarm is assigned to.

Each file must map to a valid Service Manager field. The name of the ServiceCenter/Service Manager field appears as a value of the associated `fields` subtag. For example,

```
<fields>
   <status>Status</status>
   <lastUpdate>sysmodtime</lastUpdate>
   <key>ConfigurationItem</key>
   <description>AssetTag</description>
   <assignedTo>Assignment</assignedTo>
</fields>
```

NOTE: When mapping the ServiceCenter/Service Manager fields, spaces are removed from the field name where necessary.

To verify how Service Manager fields are mapped using the above example:

1. Open up the ServiceCenter/Service Manager client and create a new connection.

   Note this example is using the Configuration Management Dashboard in the Service Manager client application.
2. Select the *Administration* perspective.
3. On the System Navigator tab collapse the *Connection* > *Favorites* and *Dashboards* > *Configuration Management* > *All PCs* node.
4. Select any one of the nodes that appears, right-click on the node and select *Open*.

   A new *Configuration Item* tab appears for the item.
5. In the Configuration Item tab, notice the available fields that can be mapped to Operations Center, including the Status, ConfigurationItem, AssetTag and Assignment fields defined in the previous example.
6. To locate the *sysmodtime* field that was mapped to the lastUpdate tag in the above example, open the *Detail Data* tab. Search for sysmodtime by using Ctrl+F in the window.

Mapping Severities

The `severities` tag predefines how ServiceCenter/Service Manager data is mapped to alarm severities.

All resulting data from the `status` tag is parsed and evaluated based on the text specified in the `severities` declarations, and an alarm status is specified any condition is met.

In the example below, if *Available*, *available*, *Installed*, *installed*, *Reserved*, *reserved*, *Transfer*, or *transfer* is found anywhere in the status data, the resulting alarm has a severity of *OK*:

```
<severities>
   <item fromRE="Available" toSeverity="OK"/>
```

```
   <item fromRE="available" toSeverity="OK"/>
   <item fromRE="Installed" toSeverity="OK"/>
   <item fromRE="installed" toSeverity="OK"/>
   <item fromRE="Reserved" toSeverity="OK"/>
   <item fromRE="reserved" toSeverity="OK"/>
   <item fromRE="Transfer" toSeverity="OK"/>
   <item fromRE="transfer" toSeverity="OK"/>
   <item fromRE="Warehouse" toSeverity="MINOR"/>
   <item fromRE="warehouse" toSeverity="MINOR"/>
   <item fromRE=".*" toSeverity="MAJOR"/>
</severities>
```

NOTE: A regular expression can be used in defining the `fromRE` attribute, as in the last declaration in the example above.

Configuring the SOAP Request

Declarations inside the `soap` tag configure the connection with the ServiceCenter/Service Manager for data requests.

For example:

```
<soap>
   <endpoint>root.soap.endpoint.prefix/SM/PWS/ConfigurationManagement.wsdl</
endpoint>
   <namespace>root.soap.namespace</namespace>
   <server>ConfigurationManagement</server>
   <port>ConfigurationManagement</port>
   <username>adapter.username</username>
   <password>adapter.password</password>
</soap>
```

The `endpoint` tag indicates the WSDL (Web Service Definition Language) file, available in ServiceCenter/Service Manager, used to map to the data we are interested in on the Service Manager Client Dashboard.

Use the endpoint tag to specify the appropriate WSDL file that contains the operations needed to retrieve the data that you want. The WSDL files map respectively to the Favorites and Dashboards items from the ServiceCenter/Service Manager Client as shown in Table 5-1 on page 187.

Table 5-1 WSDL File Mappings to the ServiceCenter/Service Manager Favorites and Dashboards

WSDL File	ServiceCenter/Service Manager Dashboard
ConfigurationManagement.wsdl	Configuration Management
ChangeManagement.wsdl	Change Management
IncidentManagement.wsdl	Incident Management
ProblemManagement.wsdl	Problem Management
ServiceLevelManagement.wsdl	Service Level Management
ServiceDesk.wsdl	Service Desk

Substitution properties, defined by property tags, contain the SOAP parameters needed to construct ServiceCenter/Service Manager Web services requests for polling or invoking operations on the ServiceCenter/Service Manager server. By default, the configuration.xml files define four of these properties. Define additional properties as required.

Table 5-2 Default Substitution Properties

Property	Sets…
root.soap.xmlns	The Soap namespace declarations required to submit Soap requests to the ServiceCenter or Service Manager server.
root.soap.envelope	The Soap envelope XML syntax.
root.soap.endpoint.prefix	The host and port of the ServiceCenter or Service Manager Soap server.
root.soap.namespace	The ServiceCenter or Service Manager Web service namespace.

Setting Up Required Operations

Operations are configured inside an `operations` tag in the `soap` tag and include both required and optional operations. The required operations are defined using the following tags:

- pollKeys
- getPollRecords

Optional operations are defined by using an operation tag and are useful in defining right-click operations on alarms that perform an action in ServiceCenter/Service Manager.

The following sections provide details on setting up mandatory operations as well as optional operations:

- "Using the pollKeys Tag" on page 188
- "Using the getPollRecords Tag" on page 189

Using the pollKeys Tag

The pollKeys tag retrieves the keys for a particular set of data. Typically, operations available for use are found in the WSDL file with names in the format of `Retrieve<something>KeysList` and are nested within a WSDL port (indicated by a `portType` tag). Often the WSDL port name is the same as the name of the WSDL file.

For example, inside the WSDL file, is the following operation:

```
<operation name="RetrieveDeviceKeysList">
```

That is nested inside of a WSDL port called ConfigurationManagement:

```
<portType name="ConfigurationManagement">
```

We can define this RetrieveDeviceKeysList operation in the configuration XML file in the name tag sub-element of the `pollKeys` tag, such as:

```
<operations>
   <pollKeys>
       <name>RetrieveDeviceKeysList</name>
       <envelope>{root.soap.envelope}</envelope>
       <body>
          <![CDATA[
             <RetrieveDeviceKeysListRequest {root.soap.xmlns}>
                 <model>
                   <keys></keys>
                    <instance>
                       <sysmodtime>&gt;={formula.poll.from.time}</sysmodtime>
                    </instance>
                 </model>
              </RetrieveDeviceKeysListRequest>
          ]]>
       </body>
       <response>&gt;RetrieveDeviceKeysListResponse</response>
        <instance>
            <container>keys</container>
            <typeattribute>type</typeattribute>
        </instance>
        <date>
            <initial>01/01/1900 00:00:00</initial>
            <format>MM/dd/yyyy HH:mm:ss</format>
        </date>
   </pollKeys>
```

```
</operations>
```

In this configuration XML example, `RetrieveDeviceKeysListRequest` is used:

- In the CDATA portion of the `body` tag and is indicated by the message attribute of the `input` tag for the definition of the `RetrieveDeviceKeysList` operation in `ConfigurationManagement.wsdl` file.
- To obtain the value of `RetrieveDeviceKeysListResponse` (note the > that precedes the `RetrieveDeviceKeysListResponse` text) and is indicated to us by the message attribute of the `output` tag for the definition of the `RetrieveDeviceKeysList` operation in `ConfigurationManagement.wsdl` file.

Note that in the CDATA section of the body tag there is the following which indicates that we are looking for all records in service manager that have a sysmodtime greater or equal to (indicated by >=) whatever is passed from the adapter as `{formula.poll.from.time}`:

```
<instance>
     <sysmodtime>&gt;={formula.poll.from.time}</sysmodtime>
 </instance>
```

The value to compare against need not be sysmodtime. However, sysmodtime is widely used in ServiceCenter/Service Manager.

Using the getPollRecords Tag

The getPollRecords tag is used to specify the operation to get the records, after the getPollRecords tag has set the operation for retrieving the keys. Typically, operations available for use are found in the WSDL file with names in the format of `Retrieve<`*`something`*`>List` and are nested within a WSDL port (indicated by a `portType` tag). Often the WSDL port name is the same as the name of the WSDL file.

For example, inside the WSDL file, is the following operation:

```
<operation name="RetrieveDeviceList">
```

That is nested inside of a WSDL port called ConfigurationManagement:

```
<portType name="ConfigurationManagement">
```

We can define this RetrieveDeviceList operation in the configuration XML file in the name tag sub-element of the `getPollRecords` tag, such as:

```
</operations>
    <getPollRecords>
          <name>RetrieveDeviceList</name>
          <envelope>{root.soap.envelope}</envelope>
          <body>
               <![CDATA[
                    <RetrieveDeviceListRequest {root.soap.xmlns}>
                         <model>
                              {formula.poll.keys}
                         </model>
```

```
                    </RetrieveDeviceListRequest>
               ]]>
          </body>
          <response>&gt;RetrieveDeviceListResponse</response>
          <instance>
               <container>instance</container>
               <typeattribute>type</typeattribute>
          </instance>
          <key>
               <container>keys</container>
               <instance>ConfigurationItem</instance>
          </key>
     </getPollRecords>
</operations>
```

Since we used the `RetrieveDeviceKeysList` operation in the previous section example, we looked in the WSDL file for an operation of the name `RetrieveDeviceList` to use for the `getPollRecords`. Following the same procedure, we reference `RetrieveDeviceList` defined under the ConfigurationManagement port, and see that `RetrieveDeviceList` goes for the name tag, `RetrieveDeviceListRequest` and `RetrieveDeviceListResponse` go for the CDATA portion and the response tags respectively.

Default Alarm Operations

By default, the configuration XML files include standard alarm operations. Because these operations call ServiceCenter/Service Manager SOAP operations, they might not be functional, depending on account permissions defined on the ServiceCenter or Service Manager server.

Table 5-3 *Default Operation Definitions in the Configuration File*

Module	Menu	Operation	Description
Change	Change Lifecycle	Approve	Approves the change ticket and updates ticket status to Approved.
		Close	Closes the change ticket and updates ticket status to Closed.
		Deny	Denies the change ticket and updates ticket status to Denied.
		Move To Next Phase	Moves the change ticket to the next step in the change process.
		Reopen	Reopens the change ticket and updates ticket status to reopened.
		Retract	Pulls the change ticket.
Inventory	Device Lifecycle	Delete	Removes the device ticket.

Module	Menu	Operation	Description
Incident	Incident Lifecycle	Close	Closes the incident ticket and updates ticket status to Closed.
		Create	Creates a new incident. Prompts the user for incident information.
			This operation can be modified to work with element property information. For more information about creating an incident prefilled with element property information, see Section 5.2.5, "Creating a ServiceCenter or Service Manager Ticket with Element Information," on page 194.
		Reopen	Reopens the closed incident ticket and updates status to Reopened.
		Resolve	Resolves the incident ticket and updates status to Resolved.
		Resolve Prompted	Resolves the incident ticket and updates status to Resolved. Prompts the user for fix type and resolution code.
	Incident Update	Assignee Info	Changes assignee information and updates status to Updated. Prompts the user for update description, assignee name, and assignment.
		Contact Name	Changes contact information and updates status to Updated. Prompts the user for update description and contact name.
Service	Call Lifecycle	Close	Closes the service ticket and changes ticket status to Closed.
		Create	Creates a change ticket. Prompts the user for change ticket information.

NO Script can be used to define alarm operations. For example, to prompt a user input. For more information about using NOC Script in configuration.xml files, see Section 5.2.4, "Defining User Prompts using NOC Script," on page 194.

Defining Custom Alarm Operations

Custom operations are defined in the configuration.xml file after required operation tags by using the `operation` and `menu` tags. They are mainly used to create right-click operations that perform an action in Operations Center and in ServiceCenter/Service Manager.

For example, in the default `ServiceManagerConfiguration.xml` file, the following operation is defined for the *Inventory* module. Defined just after the `getPollRecords` tag, it follows the same procedure as described the previous sections, and has a response tag and the CDATA section in the body tag that are used the same way as in `getPollRecords` or `pollKeys`. In addition, it uses a menu tag to indicate the sequence of menu items and sub-menus to get to perform the operation:

```
<operation name="Delete Device Record" enable="true">
  <menu>Service Manager|Device Lifecycle|Delete</menu>
  <name>DeleteDevice</name>
  <response>&gt;DeleteDeviceResponse</response>
  <envelope>{root.soap.envelope}</envelope>
  <body>
     <![CDATA[
         <DeleteDeviceRequest {root.soap.xmlns}>
```

```
            <model>
                <keys>
                    <ConfigurationItem>{alarm.ConfigurationItem}</
ConfigurationItem>
                 </keys>
                  <instance>
                  </instance>
              </model>
         </DeleteDeviceRequest>
     ]]>
  </body>
  <message>
     <container>messages</container>
     <entry>cmn:message</entry>
  </message>
</operation>
```

The `menu` tag, <menu>Service Manager|Device Lifecycle|Delete</menu>, creates a right-click operation from alarms that nests the following menu option sequence: *Service Manager* > *Device Lifecycle* > *Delete*. When *Delete* is selected, the `DeleteDevice` operation is performed on the ServiceCenter/Service Manager server.

Module XML Tags Reference

For more information about XML content and structure, reference the HP ServiceCenter or HP Service Center Soap documentation.

Table 5-4 *ServiceCenter XML Tags for Module Definition*

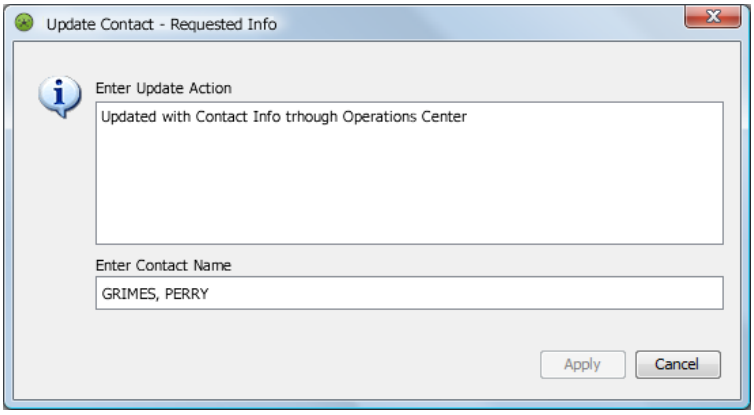
Tag	Defines…
body	The tag that wraps a Soap poll query request.
date	The initial value and format of the sysmodtime inserted for the Soap request.
endpoint	The module's Web service WSDL file URL on the ServiceCenter or Service Manager Soap servers. The Axis Soap processor uses this URL to retrieve the WSDL.
envelope	The standard envelope headers for the Soap request.
fields	The mapping of module alarm fields to Operations Center alarm columns. The following tags are required for normalization: `status` `lastUpdate` `key` `description` `assignedTo` A mapping is required for each module.

Tag	Defines…
formula	The tag that wraps `fields` and `severities` definitions.
getPollRecords	The query Soap request to retrieve the data records referenced by the keys returned by the pollKeys request. The contents are the same as those for the *pollKeys* tab.
instance	The WSDL container and data type of the data returned by the query.
name	The name of the Web service operation name as defined in the module's WSDL.
namespace	The ServiceCenter or Service Manager namespace URL for its Web service WSDL definition.
menu	Defines the menu and submenu text for the alarms operation.
operation	Web service operations exposed by the ServiceCenter or Service Manager Web services API that can be invoked for an alarm and/or element. The operation tag contains menu and prescript tags.
operations	The Web service Soap operations available for the defined module. A polling technique retrieves data from the ServiceCenter or Service Manager module. These operations are used to poll and update the Operations Center alarm display and provide additional Web service operations for alarms and elements. The operations tag contains `pollKeys`, `getPollRecords`, and `operation` tags.
password	The associated password for ServiceCenter's or Service Manager's Web service authentication security requirements.
pollKeys	A query to obtain a list of keys of module records changed since the last poll time. The initial query retrieves all module records contained in the ServiceCenter or Service Manager database. Subsequent queries use the sysmodtime to retrieve only those records that changed since the last poll query. The pollKeys tag contains name, envelope, body, response, instance, and date tags to generate a Soap request that executes the query and to describe the Soap response for this query.
port	The tag name within the associated WSDL file for the Web services Soap port.
prescript	A NOC Script segment to execute before sending the generated Soap request to the ServiceCenter or Service Manager Soap server.
properties	Each module tag requires one properties tag definition. Each properties tag must contain a `formula` and `soap` tag definition.
response	The WSDL response tag for this operation.
server	The tag name within the associated WSDL file for the Web services Soap server.
severities	The mapping of the defined Modules status to Operations Center' alarm severities.
soap	Contains `endpoint`, `namespace`, `server`, `port`, `username`, `password`, and `operations` tags required to define a Soap request and response.
username	The user name and associated password for ServiceCenter's or Service Manager's Web service authentication security requirements.

5.2.4 Defining User Prompts using NOC Script

Use NOC Script to prompt for input before sending Soap operation requests to the ServiceCenter or Service Manager Soap server. Define any necessary NOC Script using the operation and prescript tags in the associated configuration XML file.

Figure 5-4 *The Custom User Prompt Dialog box for Incident Update Contact Name Operation*



When using NOC Script in a configuration XML file, the NOC Script code must be inside a `CDATA` delimiter within the `prescript` tag.

By default, the configuration XML files defines several user prompts as part of operation definitions.

For more information about NOC Script, see the *Operations Center 5.5 Scripting Guide*.

5.2.5 Creating a ServiceCenter or Service Manager Ticket with Element Information

Use NOC Script to retrieve property information from any element and use it to create a ServiceCenter Incident ticket.

In the following implementation, we modify the Create Incident script normally used for alarms and create a *Create Incident* operation for elements that allows users to create a new incident record in the ServiceCenter or Service Manager server for any element in the Elements hierarchy.

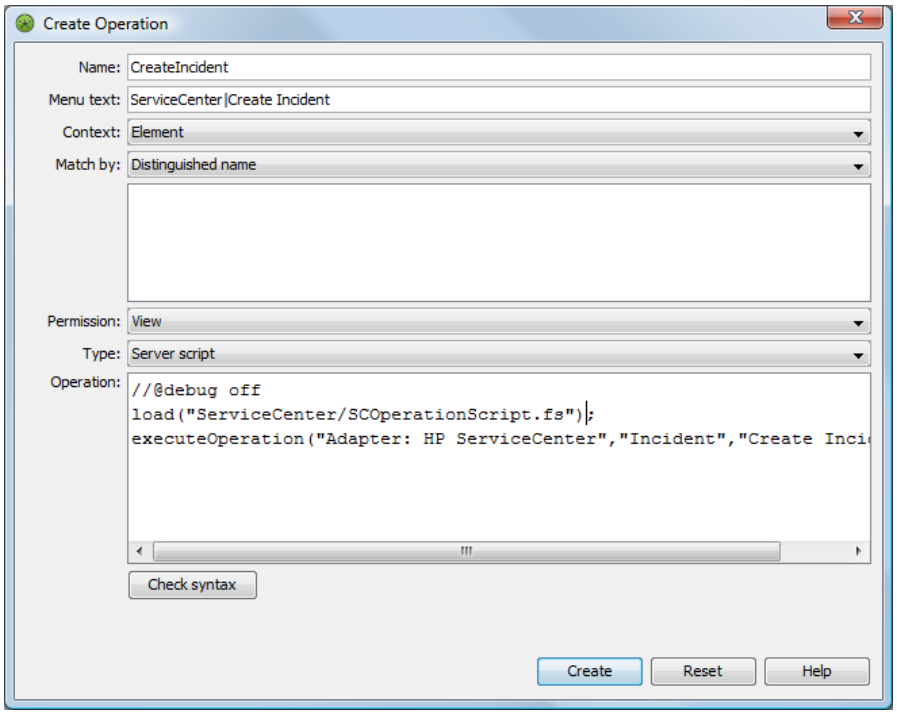
For more information about the default Create Incident operation definition, see Table 5-3, "Default Operation Definitions in the Configuration File," on page 190)

To create a custom operation to create ServiceCenter/Service Manager tickets:

1. Create an operation definition that is applicable to all Operations Center elements that run a NOC Script.

   For this implementation, the default `Create Incident` operation from the configuration XML file can be modified (or reused to create a new script) by updating the property references for element properties which are always prefixed with `element.` instead of an `alarm.` prefix.

   For more information about defining custom operations, see the *Operations Center 5.5 Server Configuration Guide*.
2. Access the operation definition in the *Explorer* pane under *Administration* > *Server* > *Operation Definitions*.
3. Right-click *Operation Definitions*, then select *Create Operation*.



4 Enter the following code in the *Operation* field to call the `SCOperationScript.fs` file found in the */OperationsCenter_install_path*`/database/scripts/ServiceCenter` or */OperationsCenter_install_path*`/database/scripts/ServiceManager` directory:

```
load("ServiceCenter/SCOperationScript.fs");
```

The SCOperationScript.fs script locates the ServiceCenter or Service Manager adapter and invokes the module operation. An operation prescript usually detects whether the operation was invoked from an Element or Alarm and populates the ServiceCenter Soap request with appropriate information. The script can be used with IDK and non-IDK elements.

5 Enter the following code in the *Operation* field to call a default operation as defined in the `ServiceCenterConfiguration.xml` or `ServiceManagerConfiguration.xml` file (or `ServiceManagerConfiguration_9.3.xml` (for Service Manager 9.3)), then click *Apply*:

- For HP ServiceCenter enter:

  ```
  executeOperation("Adapter: HP ServiceCenter(r)","Incident","Create
  Incident");
  ```

- For HP Service Manager enter:

  ```
  executeOperation("Adapter: HP Service Manager(r)","Incident","Create
  Incident");
  ```

The first argument is the name of the adapter as displayed in Operations Center. The second argument is the module name as specified in the module tag. The third argument is the operation name as specified in the operation tag.

When this operation is performed on an element, the *Create Incident* operation opens a *Create Incident* dialog and pre-fills element property information as required.

6 Click *Create*.

6 Using Remote Containers

A Remote Container is used to distribute the running of Operations Center adapters and integrations across several machines or to run Operations Center adapters and integrations under a different JVM, or other configurations, than the Operations Center server.

A Remote Container server is a Java program that runs in its own Java Virtual Machine (JVM) on either the same host machine or a different host machine as a Operations Center server, and uses the same daemon port as the Operations Center server.

Use Remote Containers when you want to:

- **Tighten Security on Firewall:** If running the Operations Center server outside your firewall and you need to limit the number of open ports to the firewall, install a Remote Container on a host server inside the firewall that will connect with all your management systems and integrations inside the firewall. The Remote Container connection requires only one port to be open to deliver all information back to the Operations Center server.
- **Integrate to Different JVMs:** If running an Operations Center server with a 64-bit JVM, configure the Remote Container to use a 32-bit JVM in order to integrate to Remedy 7, which is an integration that only supports 32-bit JVMs.The Remote Container connects to the Remedy integration and reports back to the Operations Center server.
- **Integrate to Different Operating Systems:** If the Operations Center server is running on a UNIX server and you need to integrate to SMARTS EMC (which is only supported on Windows), install a Remote Container on the Windows server. The Remote Container connects to the SMARTS EMC integration and reports back to the Operations Center server.
- **Decrease Memory Demands and Start Up Time:** If the Operations Center server is at memory capacity, you can off-load one or more running adapters and integrations onto Remote Container servers running on a different host server. In addition, off-loading adapters and integrations can improve Operations Center server start up time.

Adapter instances on a Remote Container server are configured and viewed using the Operations Center console. Elements originating from adapters on Remote Container servers transparently display alongside local elements. A Remote Container adapter is like any other Operations Center adapter that collects information from management systems. The difference is a Remote Container adapter collects information from the Remote Container server.

For more information about Remote Containers and how they are different from an Operations Center server, see Section 6.1, “About Remote Containers,” on page 198.

The basic steps to setting up and using Remote Containers are:

1. **Install the Remote Container server.** A Remote Container server is automatically installed with Operations Center. You’ll only need to do this step if you want to run the Remote Container on a different host server from Operations Center, or create additional Remote Container instances than those defined by default.

   For instructions, see Section 6.2, “Installing Remote Container Servers,” on page 201.

2. **Configure and Manage the Remote Container servers.** Remote Container server options are set in the Operations Center Configuration Manager and apply to all Remote Containers.

   For instructions, see Section 6.3, “Configuring Remote Container Servers,” on page 202 and Section 6.4, “Starting, Monitoring or Stopping a Remote Container Server,” on page 205.

3. **Define connections to the Remote Container servers.** One or more connections must be created to allow Operations Center to communicate with the Remote Container. They also run all adapters and integrations just like in Operations Center.

   For instructions, see Section 6.5, “Defining Remote Container Connections,” on page 206.

4. **Create adapters on the Remote Container.**

   For instructions, see Section 6.6, “Configuring Adapters on Remote Containers,” on page 209.

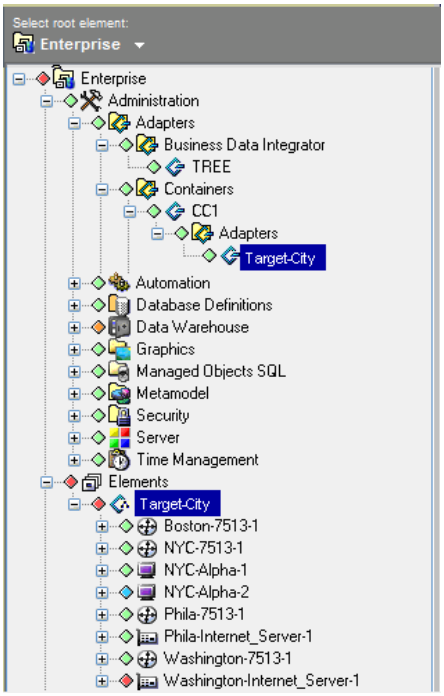
6.1 About Remote Containers

A Remote Container server is a Java program that runs in its own Java Virtual Machine (JVM) on either the same host machine or a different host machine as a Operations Center server. There is no limit on the number of concurrently running Remote Container servers.

A Remote Container server is similar to a Operations Center server with regard to running adapters and integrations, but it does not have any other Operations Center features (such as Service Models, BSLM, Data Warehousing, MOSQL, ACLs, and so on).

The adapters and integrations running on a Remote Container server display under the local Operations Center server Elements branch and appear to be local elements. The full set of Operations Center server features apply to these elements as if they were local elements.

Figure 6-1 *The Target-City adapter has started on the CC1 Remote Container server. The Target-City adapter and its child elements display in the local Elements branch*



The following topics describe specific attributes and differences of the Remote Container implementation:

6.1.1 Supported Adapters and Integrations

Note that the following adapter types are not certified to run in Remote Containers:

- Event Manager
- InterConnection (ICA)
- Data Integrator: The creation or editing of Data Integrator definitions from a Remote Container are not supported. Only the running of deployed Data Integrator Adapters is supported.

6.1.2 Adapter Communications

A Remote Container adapter communicates with a Remote Container server by requesting a reference through the Remote Container server's associated daemon process. When a Remote Container server starts, its name is registered with its daemon process, which allows a Remote Container adapter to contact the Remote Container server and establish a connection. This implies that a Remote Container adapter cannot establish a connection unless the named Remote Container server has started and registered its name with the daemon.

The communications protocol is CORBA-based and relies on the same communications mechanisms as the Operations Center ICA adapter (also known as F2F). ICA adapter behavior is similar to that of the Remote Container adapter. The Remote Container adapter behavior differs from the ICA in that the Remote Container adapter appears as a locally running adapter.

6.1.3 Adapter Name Conflicts

One of the advantages of using Remote Containers is that element dnames appear local to the server regardless of where the adapter is actually running.

However, it is possible to configure adapter definitions with the same name on more than one server, such as to allow fail-over and remain transparent to the user. This causes duplicate adapter definition names to exist on different Remote Container servers, which can conflict with one another or with the names of adapters defined on the local Operations Center server.

By design, when duplicates exist, only the first recognized adapter instance displays under the local Operations Center server *Elements* branch; and an error message is logged in the `formula.trc log` file when the adapter with the duplicate name is started:

`Could not display Remote Container Adapter` *`Adapter_Name`* `Container` *`RemoteContainer_Name`* `due to an already running Adapter with that name.`

For example, if the Remote Container servers `ContainerRCServer1` and `ContainerRCServer2` both have an adapter named `SystemE`, and the local Operations Center server also has an adapter named `SystemE`, then only one instance can display under the local Operations Center server *Elements* branch. The first instance that is started becomes the adapter shown in the *Elements* branch, and no additional occurrences display.

6.1.4 Directory Structure

The Remote Container server directory structure is the same as a Operations Center server installation directory structure. This allows for ease of administration, configuration, maintenance, and patching. The Remote Container server has `/bin`, `/classes`, `/config`, `/database`, `/integrations`, `/logs`, and `/patches` directories that are used in the same way as a Operations Center server.

The differences are:

- In the `/config` directory, the custom properties file uses the name `Container`*`RemoteContainer_name`*`.custom.properties` instead of `Formula.custom.properties`.

  For example, for a Remote Container server named RCServer1, the custom properties file is named `ContainerRCServer1.custom.properties`.

  If this file exists in the `/config` directory, then the properties within are used by the RCServer1 Remote Container server and adds to or overrides properties defined in the `Formula.properties` file.

  For more information about custom properties files, see "Making Custom Changes" in the *Operations Center 5.5 Server Configuration Guide*
- In the `/logs` directory, a running Remote Container server logs to a file named `Container`*`RemoteContainer_name`*`.trc`.

  For more information, see "Log Messages" on page 201.
- When a Remote Contain server starts, it creates a directory tree under `/`*`OperationsCenter_install_path`*`/containers/`*`RemoteContainer_name`* if the directory does not already exist.

  For example, for a Remote Container server named RCServer1, a directory `/`*`OperationsCenter_install_path`*`/containers/ContainerRCServer1` directory is created.

  Under the *`RemoteContainer_name`* directory, there are two subdirectories:

  1. A `/configstore` directory holds the persisted configuration information for the Remote Container server.
  2. A `/database` directory holds configstore-related files.

  Do not place new files or modify the files in these directories. Usually, the files in these directories do not need to be patched or modified.

It is recommended that you back up these files as you do other Operations Center directories.

6.1.5 Web Server and Clients

A Remote Container server does not start an associated Web server and does not allow client sessions. The only connection allowed to a Remote Container server is through a Remote Container adapter instance on a Operations Center server.

6.1.6 Log Messages

Log messages for a running Remote Container server are written to the `/OperationsCenter_install_path/logs/ContainerRemoteContainer_Name.trc` file.

For example, for a Remote Container server named `RCServer1`, the log file is named `ContainerRCServer1.trc`. All log messages for the Remote Container server RCServer1 are written to this file. The file name and logging level can be changed in the *Remote Container* section of the Operations Center Configuration Manager.

For more information about error messages that are logged when there are adapter name conflicts, see Section 6.1.3, "Adapter Name Conflicts," on page 199.

6.1.7 Patches

Operations Center patch bundles are used to supply any updates for Remote Container servers.

6.2 Installing Remote Container Servers

Remote Container servers can be installed and configured in the following ways:

- **Running Multiple Servers on the Same Host:** Run one or more servers on the same host and off the same daemon as an Operations Center server. This enables an administrator to off-load one or more running adapters and integrations on one or more Remote Container servers, each running in a separate Java Virtual Machine, without having a second installation of Operations Center on the same host machine.
- **Using a Dedicated Remote Container Server Host:** This installation method uses a Operations Center installation that is dedicated only to running only Remote Container servers and does not run a Operations Center server. Multiple Remote Container servers can run on a remote host machine separate from the Operations Center server host machine.

The system requirements for a Remote Container server are the same as for a Operations Center server. For more information, see the *Operations Center 5.5 Getting Started Guide*.

Remote Container is always installed when the Operations Center is installed. However, it can be installed separately by using the custom installation option in the Operations Center installer.

To install the Remote Container and create Remote Container instances:

1. If using a dedicated host from the Operations Center server, install the Remote Container server using the custom installation option in the Operations Center installer.

   For more information, see the *Operations Center 5.5 Server Installation Guide*.
2. Installation automatically includes the two Remote Container instances that are predefined as `ContainerRCServer1` and `ContainerRCServer2`.

   To define a new Remote Container instance, issue the following command:

   `mkcontainer RemoteContainer_name`

   For example, use the `mkcontainer SystemE` command to create a Remote Container server named `SystemE`.
3. Continue to Section 6.3.1, "Setting Configuration Options for Remote Containers," on page 202 to enable the new Remote Container server instances.

6.3 Configuring Remote Container Servers

6.3.1 Setting Configuration Options for Remote Containers

After installing a Remote Container server either with the standard Operations Center install or separately as a custom install, Remote Container configuration options must be set using the Operations Center Configuration Manager.

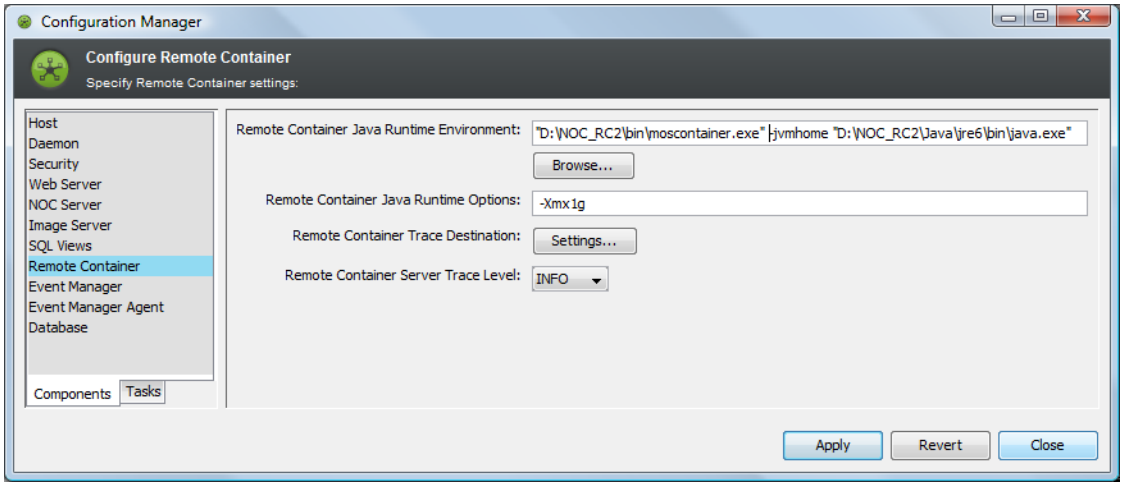
Since any number of Remote Container servers can run from a single installation, updating the Configuration Manager settings updates the settings for all Remote Container servers that have been defined.

To configure Remote Container servers:

1. Open the Configuration Manager by performing one of the following steps, depending on your operating system:
   - For Windows, from the desktop, click *Start* > *Programs* > *NetIQ Operations Center* > *Configure NetIQ Operations Center*.
   - For UNIX, from the `/OperationsCenter_install_path/bin` directory, enter `Customizer` at the command prompt.

   For more information about the Configuration Manager, see the *Operations Center 5.5 Server Configuration Guide*.
2. Click *Remote Container*.

   The Remote Container server settings display in the right pane:



3 Configure the following settings as necessary.

Setting	Description
Remote Container Java Runtime Environment	Specifies the path to the `moscontainer.exe` file. The `-jvmhome` setting points to the Java Runtime Environment (JRE) used to run the Remote Containers servers.
Remote Container Java Runtime Options	Command line options used by the specified JRE for running the Remote Container server. Usually does not need to be changed.
Remote Container Trace Destination	The settings for the Remote Container server trace logs.Click *Settings* and configure the logging TRC file output. For more information about configuring Trace Log settings, see "Configuring Trace Logs" in the *Operations Center 5.5 Server Configuration Guide*.
Remote Container Server Trace Level	Controls how much information is passed to the Remote Container server trace logs. Set the trace file logging level. For more information about configuring Trace Log settings, see "Configuring Trace Logs" in the *Operations Center 5.5 Server Configuration Guide*.

4 To configure the remote container to start when the daemon starts, do the following:

4a Click *Daemon*.

4b If only Remote Container servers run from this installation, then delete the default `Database Image Formula` value from the *Automatically Start Servers* setting.

4c Append the *Automatically Start Servers* setting with the names of the remote container servers to that should start automatically when the mosdaemon starts.

For example, `ContainerRCServer1,ContainerRCServer2`

By default, two predefined Remote Container servers are installed: (`ContainerRCServer1` and `ContainerRCServer2`). Additional Remote Container servers can be created using the `mkcontainer` command (see Step 2 on page 201).

5 Restart the Remote Container server.

6.3.2 Customizing Remote Container Servers for Data Integrator Adapters

Data Integrator definitions cannot be created on a Remote Container; they must be created and deployed on a Operations Center server. Data Integrator definitions are only available to Remote Containers as adapter instances.

If running a server that is dedicated to Remote Containers and not sharing the same directory structure as the Operations Center server, you must manually copy the JAR file for Data Integrator integrations to the Remote Container server before the adapter instance can run on the Remote Container server.

NOTE: For an installation having both a Remote Containers and a full Operations Center integration sharing the same directory structure, you do not need to perform the following procedure.

To prepare a Remote Container server (not sharing the same directory structure as the Operations Center server) to run Data Integrator adapters:

1. After deploying the Data Integrator definition, copy the *DataIntegratorDefintionName*.jar file from the /*OperationsCenter_install_path*/integrations directory on the Operations Center server to the */OperationsCenter_install_path*/integrations directory on the Remote Container server.

   For example, if you have created and deployed a Data Integrator definition named DBDATA on server A and wish to create and run an adapter instance on an installation of Remote Containers on server B, copy */OperationsCenter_install_path*/integrations/*DBDATA*.jar from server A to the */OperationsCenter_install_path*/integrations directory on server B.
2. Restart the Remote Container.

For more information about the Data Integrator, see the *Operations Center 5.5 Data Integrator Guide*.

6.3.3 Changing Individual Remote Container Server Settings

To configure a specific Remote Container server without affecting other Remote Container servers that have been defined, you must manually edit the Remote Container's INI file or add a custom properties file for the Remote Container.

To make setting changes for a single Remote Container server:

1 Do one of the following:

- Edit the INI file for the remote container, */OperationsCenter_install_path*/config/template/Container*RemoteContainer_name*.ini

  For example, ContainerRCServer1.ini is the INI file for a Remote Container server named *RCServer1*.
- Create and add the custom setting to a custom properties file named Container*RemoteContainer_name*.custom.properties in the */OperationsCenter_install_path*/config directory.

  For example, ContainerRCServer1.custom.properties is a properties file for a Remote Container server named RCServer1.

  For more information about custom properties files, see "Making Custom Changes" in the *Operations Center 5.5 Server Configuration Guide*

2 Stop and restart the Remote Container server for changes to take effect.

For instructions, continue to Section 6.4, "Starting, Monitoring or Stopping a Remote Container Server," on page 205.

6.4 Starting, Monitoring or Stopping a Remote Container Server

The following sections provide information about the available ways to start, stop or check status for a Remote Container server:

- Section 6.4.1, "Automatically Starting, Stopping and Checking Status of Remote Containers with the Operations Center Daemon," on page 205
- Section 6.4.2, "Starting, Monitoring or Stopping a Remote Container Server from the Command Prompt," on page 205
- Section 6.4.3, "Managing the Remote Container Server from the Operations Center Console," on page 206

6.4.1 Automatically Starting, Stopping and Checking Status of Remote Containers with the Operations Center Daemon

Use the *Automatically Start Servers* setting in the Configuration Manager to configure the Remote Container to start, stop and display status when you issue commands to the Operations Center daemon. For instructions, see Step 4 on page 203.

6.4.2 Starting, Monitoring or Stopping a Remote Container Server from the Command Prompt

Use the same start, monitor status, and stop commands for the Remote Container that you do to administer the Operations Center server.

To start, monitor or stop a Remote Container server:

1 Log in as the user `formula` (or any user with root privileges) and from the `/OperationsCenter_install_path/bin` directory, issue the appropriate command as outlined in Table 6-1.

Table 6-1 *Remote Server Commands*

Action on Remote Container	Issue Command
Start Remote Container Server	`mosstart RemoteContainer_name` where *container_name* is the name of the Remote Container server. For example, if the Remote Container name is ContainerRCServer1: `mosstart ContainerRCServer1`
View Runtime Status	`mosstatus RemoteContainer_name`
Stop Server	`mosstop RemoteContainer_name`
Stop All Servers for the Daemon	`mosstop –shutdown`

6.4.3 Managing the Remote Container Server from the Operations Center Console

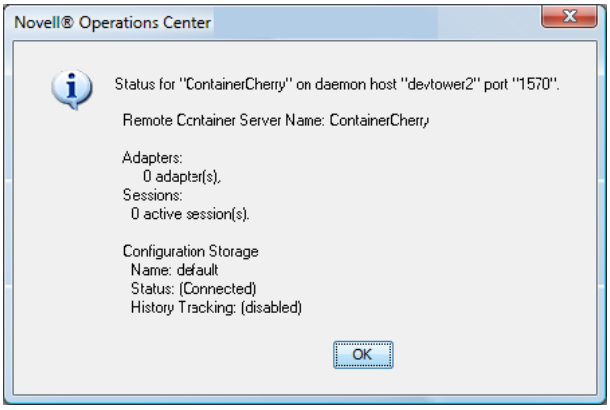
Use the right-click options on a remote container connection to start, stop, and monitor the associated Remote Container servers.

Container Server operations contact the daemon on the host and port specified for the container connection and determine if the Remote Container server named in the container connection definition(s) is registered with the daemon.

For instructions on creating connections to a Remote Container, see Section 6.5, "Defining Remote Container Connections," on page 206.

To start, stop and display status for a Remote Container server:

1. In the *Explorer* pane, click *Administration* > *Adapters* > *Containers*.
2. Right-click an associated connection, click *Container Server* and select one of the following:
   - *Start Container Server*
   - *Stop Container Server*
   - *Status Container Server*. If the Remote Container server is registered with the daemon, a status dialog box displays.

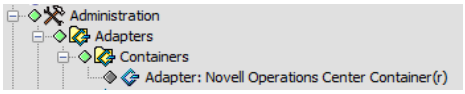


   - *Check Container Server*. Checks to see if the Remote Container server is running.
   - *List Container Servers*

6.5 Defining Remote Container Connections

After configuring a Remote Container server, define a connection between a Operations Center server and a Remote Container server. When you create a container connection, you are, in fact, defining an instance of a Remote Container adapter (see Section 6.6, "Configuring Adapters on Remote Containers," on page 209).

Container connections display beneath the *Containers* element in the *Explorer* pane.



The following topics describe how to create and manage Remote Container connections:

- Section 6.5.1, "Defining Connections to the Remote Container," on page 207
- Section 6.5.2, "Managing Server Connections," on page 208

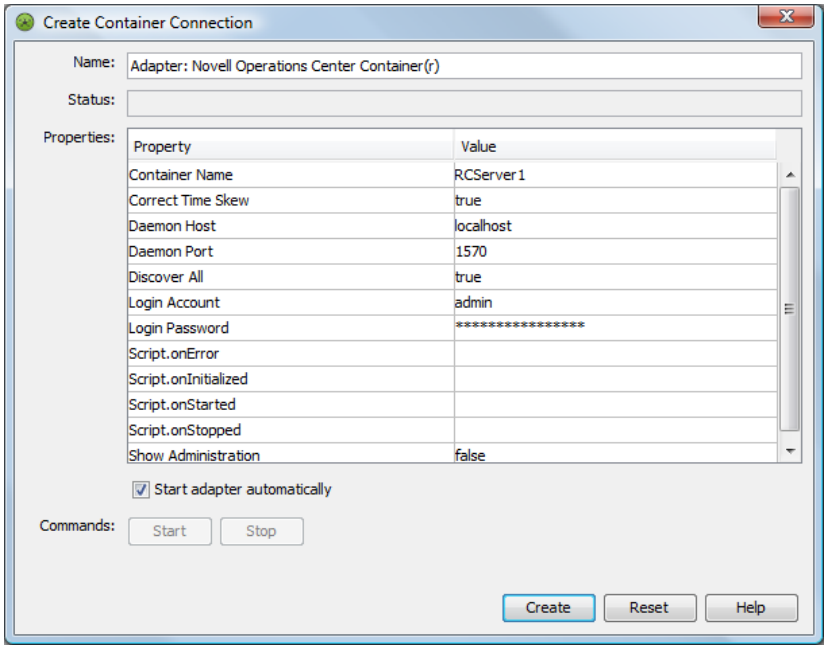
6.5.1 Defining Connections to the Remote Container

If you define the connection between the Operations Center server and the Remote Container server to use full security (such as SSL communication), then Operations Center validates the certificate dates, particularly checking for expired or invalid certificate dates.

To define a Remote Container server connection:

1. In the *Explorer* pane, click *Administration* > *Adapters*.
2. Right-click *Containers*, then click *Create Container Connection*.

   The Create Container Connection dialog box displays.



3. Update the fields as required.

Setting	Description
Container Name	The name of the Remote Container server to connect to the Managed Object server. The default is `RCServer1`.
Correct Time Skew	Corrects the time skew difference between the Operations Center server on which this container connection is defined and the specified Remote Container server. Set to `false` to perform no corrections.The default is `true`.
Daemon Host	The host name of the Remote Container server daemon. The default is `localhost`.
Daemon Port	The port number of the Remote Container server daemon. The default is `1570`.

Setting	Description
Discover All	Force discovery of all of the remote elements. Set to `false` to discover as needed. The default is `true`.
Login Account	The login account for the Remote Container server. The default is `admin`.
Login Password	The login password for the Remote Container server. The default is `formula`.
Show Administration	When set to `false`, only *Adapter* elements display under the container connection element. Set to `true` to display *Administration* and *Adapter* elements from the Remote Container server as children of the container connection element, when the connection is started. The default is `false`. The *Administration* branch for Remote Containers is limited to the *Adapters*, *User* and *Group Security*, and *Sessions* branches. You cannot assign access control on a Remote Container server, so it is recommended that the *Security* branch be used only for changing the admin password, as needed.

4 Click *Create*.

5 Continue to Section 6.5.2, "Managing Server Connections," on page 208 for instructions on starting or verifying the connection, or continue to Section 6.6, "Configuring Adapters on Remote Containers," on page 209 for instructions on creating adapters instances to run on the Remote Container.

6.5.2 Managing Server Connections

Use right-click options to manage all server connections or only a specific connection, depending on the element from which the option is issued:

- "Start, Stop, Delete Container Connections" on page 208
- "Edit Container Connection Properties" on page 209

Start, Stop, Delete Container Connections

When a container connection starts and a connection is made, the Administration/Adapters element from the Remote Container server displays beneath the container connection element. However, if the Show Administration container connection property is `True`, then the entire Administration branch displays beneath the container connection element.

Elements received through the adapters that run on the Remote Container server display under the local Operations Center server *Elements* branch.

To start or stop Remote Container server connections, do any of the following:

1 To perform an action on all remote container instances, right-click the *Containers* element, and do one of the following:

- Click *Start all Container Connections*.

  This option is available only if there are container connections that are not started.

- Click *Stop all Container Connections*.

  This option is available only if there are running container connections.

2 To perform an action on a single remote container, do any of the following:

- Click *Start Container Connection*.

This option is available only if the container is not started.

- Click *Stop Container Connection*.

  This option is available if the container connection is running.

- Click *Delete Container Connection*, and click *Yes* in the confirmation box to delete the container connection.

Edit Container Connection Properties

To edit the properties of a Remote Container server connection:

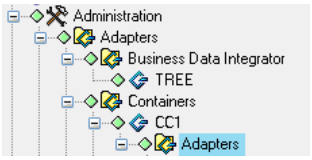
1. Right-click a container connection element, then click *Properties*.
2. In the left pane, click *Container*.
3. Edit the properties and then click *Apply*.

6.6 Configuring Adapters on Remote Containers

An adapter running on a Remote Container is like any other Operations Center adapter that collects information from management systems for use on a Operations Center server. However, the key difference is a Remote Container adapter collects information from a Remote Container server for use in Operations Center.

When you create a container connection (see Section 6.5, “Defining Remote Container Connections,” on page 206), you are, in fact, defining an instance of a Remote Container adapter. It is from these connection adapters that you can define any adapter and integrations you want to run from the Remote Container.

Figure 6-2 *The CC1 adapter is a connection to a Remote Container server*



To create an adapter on a Remote Container:

1. Verify ORB software has been installed, if required, and any additional integration and configuration steps have been performed.

   For requirements and instructions, refer to the specific integration section in:

   - Chapter 3, “Application and Management System Integrations,” on page 23
   - Chapter 4, “Discovery Tool Integrations,” on page 143
   - Chapter 5, “Trouble Ticket Systems Integrations,” on page 165
2. In the *Explorer* pane, click *Administration* > *Adapters* > *Containers* > `Connection_Name`.
3. Right-click *Adapters* and select *Create Adapter*.
4. Define the adapter definition and specify adapter properties.

   For general information about creating adapters, see Chapter 2, “Creating Adapters,” on page 17. For information about integration property settings, see Appendix A, “Adapter Property Reference,” on page 279.

7 NOC Universal Adapter

Using the Operations Center Universal Adapter, you can take raw data from multiple sources, such as log files and databases, and represent it as events in Operations Center. These data sources might include financial, help desk tracking, and customer-related data or data from management systems that do not directly integrate to Operations Center.

Integrating with the NOC Universal Adapter requires custom scripting to perform actions such as clear, close, update, and so on.

This section is intended for administrators who create Operations Center scripts for use with the NOC Universal adapter. It assumes that you know how to use NOC Script commands to customize Operations Center enterprise management capabilities. Sample scripts are shown with brief explanations. It is assumed that the person who implements these types of scripts has an excellent understanding of Java scripting.

The following section describe how to setup the NOC Universal Adapter:

7.1 Introduction

The Universal Adapter allows you to create events from multiple sources using databases or log files and be able to define new field/value pairs on the fly.

Data integration methods include: Perl, SQL scripts, C++, Java Script, compiled Java, VB or REXX. The choice depends on the optimal way to obtain data and deliver it to Operations Center. In each case, there are many ways of configuring the data that the Operations Center server receives, such as using timed polling intervals or using the actual data source to push the data. For example,

- Create a Perl script to monitor the end of a log file and send updates to Operations Center as new data arrives.
- Place a trigger method in the database to send the stream of data to the Operations Center server as it arrives.

While the Event Manager also gathers data from log files and represents data as events in Operations Center, it uses rule sets that define the log file and assign variables and values together. Out of the box, Event Manager can clear alarms, assign alarms and perform other useful operations and correlation incremental counters. If using the NOC Universal adapter, it is necessary to develop these operations.

7.2 Configuring the NOC Universal Adapter

The NOC Universal adapter listens on a specific port on the Operations Center server for a stream of text that has specific formatting requirements. It is possible to create field names as needed, but there are some required fields. If alarms are sent via data streams, there is a required beginning string, ending string and a few required fields. The incoming stream of data is represented in Operations Center as an event.

Alternatively, connect a script to the adapter, mine the data source and call the Operations Center createAlarm function directly to create alarms and events in Operations Center.

Since this is a manual integration with a back-end system, there is the option of adding custom operations that can perform actions such as clear, close, update, and so on. Each of these operations is dependant on the back-end system. For instance, if the data originates from a back-end database that tracks help desk tickets, consider adding an operation such as Close Ticket. This operation prompts the user for input and then issues an SQL update against the database.

The following are general steps with details explained in subsequent sections.

To set up and use the NOC Universal adapter:

1. Verify the incoming data stream has required fields and correct formatting.

   For information about data streams, see Section 7.3, “Setting Up the Incoming Data Stream,” on page 213.
2. Select a method for bringing data into Operations Center.

   If using databases to update Operations Center, add JDBC drivers to a Operations Center server.
3. Create a Operations Center script.

   Use NOC Script commands to create alarms and events. Add custom operations such as *Clear*, *Close*, *Update* (optional).
4. Modify the `scripthierarchy.xml` file for hierarchical displays.
5. In Operations Center, create a new NOC Universal adapter (NOC Script) adapter for each host or data source. For instructions, see Section 2.1, “Creating an Adapter,” on page 17.
6. Modify adapter properties.

   For instructions, see Section A.32, “NetIQ Operations Center Universal,” on page 339.
7. Start the adapter.

   For instructions, see Section 2.2.1, “Starting, Stopping, or Deleting an Adapter,” on page 18.
8. If using a log file to update Operations Center, parse the data in the data source.
9. Gather and send data to a server on a specific TCP/IP port.

7.3 Setting Up the Incoming Data Stream

The following is an example of a fully-formatted stream of data that displays as a Operations Center event when sent to the NOC Universal adapter:

Component	Example Data Stream
Stream Header (Required)	`### EVENT ###` `Database_Monitor;`
Required Data Fields	`originating_event_id="34532";` `msg="Instance is no longer active, tnsping failed";` `severity="CRITICAL";`
Additional Data Fields	`hostname="server45.mosol.com";` `process="oracle";` `instance="ORCL";` `application="people soft";` `testing_id="12345";`
Stream Footer (Required)	`END` `### END EVENT ###` `PROCESSED`

The following sections explain components of the data stream and how to test the validity of the data stream:

- Section 7.3.1, “Required Stream Header and Event Class,” on page 213
- Section 7.3.2, “Data Fields,” on page 214
- Section 7.3.3, “Required Stream Footer,” on page 215
- Section 7.3.4, “Closing the Event by Data Stream,” on page 215
- Section 7.3.5, “Manipulating the Event Time Stamp,” on page 215
- Section 7.3.6, “Testing the Validity of the Data Stream,” on page 215

7.3.1 Required Stream Header and Event Class

All streams of data (events) must start with the following text on a line by itself:

```
### EVENT ###
event_class;
```

Where *event_class* is an identifier that represents an event class. For example, if the information source is a log file generated by a product named `MonitorMyWWW` that monitors Web sites. Replace *event_class* with `MonitorMyWWW`.

During the initial sending of data, the event_class value is set to the main class of the event. For data updates, this field is set to Sync to update existing events. In cases where there is no initial data mining capability, all events are sent as Sync and the adapter property `EventConsoleName` assigns the event class value. The event_class value is also useful defining right-click operations in the implementation.

7.3.2 Data Fields

Each incoming data field is represented by a field name and value, as shown in the following example:

```
Host="taz";process="oracle";instance="ORCL";
```

Requirements for data fields are:

- Separate each field/value pair with a semicolon.
- Fields are case-sensitive in Operations Center.
- `originating_event_id`, `severity`, and `msg` are required data fields. They are mandatory for Operations Center to be able to process the data stream correctly.

Required Field Name	Descriptions
originating_event_id	A unique identifier that allows correct application of custom operations (such as clear or close) and updating of events later. ◆ Value can contain numbers and letters. ◆ If the system's unique ID contains characters, it might be necessary to implement a hashing scheme to convert it to a number.
severity	A severity level for the event. ◆ Value must be must be all uppercase and must contain one of these valid strings: ◆ **CRITICAL** display color is red. ◆ **MAJOR** display color is orange. ◆ **MINOR** display color is yellow. ◆ **INFORMATIONAL** display color is blue. ◆ **OK** display color is green. ◆ **UNKOWN** display color is gray. ◆ If the severity field is set incorrectly, it defaults to `UNKNOWN`. Set unknown severity to UNKNOWN or pass codes other than the ones listed above.
msg	A description about the event which can include the cause, or information that displays after opening an event or moving the mouse over an event in the Operations Center *Network* view. ◆ Value can contain numbers, characters, and any standard symbol, except semicolons. ◆ If the value itself contains quotes, be careful not to break or mismatch the quotes as the event might process incorrectly.

- `originating_tec_hostname` is not required for valid events. However, it must be included to have right-clicks for Close and Acknowledge work successfully. See Section 7.6, "Understanding Alarm Operations and Event Status," on page 217.
- Data can be sent using multiple lines, such as:

```
Host="taz";
Process="oracle";
Instance="ORCL";
```

- If the event is sent in one line, the last field/value pair must have a carriage return embedded in the stream just before the ending footer data.
- Surrounding values with quotation marks is optional, unless the value contains spaces.

7.3.3 Required Stream Footer

The end of the event must end with the following text on separate lines:

```
END
### END EVENT ###
PROCESSED
```

7.3.4 Closing the Event by Data Stream

When closing an event, it is necessary to specify the event class and define `originating_event_id` and `status`.

An example event stream to pass in order to close the alarm:

```
### EVENT ###
Database_Monitor;
originating_event_id="34532";
status="CLOSED";
END
### END EVENT ###
PROCESSED
```

7.3.5 Manipulating the Event Time Stamp

The date and time of the event creation/occurrence is the date and time when the NOC Universal adapter received the event. It has no relation to any date or time contained within the event.

You can use the `date_reception` field name to specify a specific date/time stamp.

To specify a hard-coded event time stamp:

1. Stop the adapter.
2. Right-click the adapter, and select *Properties*.
3. Enter `date_reception` as the value for the *UpdateTimestamp* property.
4. Add the following tag after the stream header tag:

   date_reception="*unix_timestamp*"

   To convert the desired date/time to a Unix time stamp, use a Web site such as OnlineConversion.com.

7.3.6 Testing the Validity of the Data Stream

To test the validity of the data stream:

1. Telnet to the port on which the NOC Universal adapter is listening.
2. Send the sample data stream.

   An event is created for the adapter.

7.4 Creating Log Files for the NOC Universal Adapter

Create separate log files for each script adapter by modifying the `formula.properties` file. The logging feature uses the log4j interface and is controlled using the standard `formula.properties` entries for logging.

In this example, logging is controlled for a script adapter named `feb-webspective`.

To enable a script adapter to log incoming events:

1. Add the following lines to the `formula.properties` file:

```
#
log4j.appender.wst=org.apache.log4j.RollingFileAppender
log4j.appender.wst.File=../logs/webspective.rl
log4j.appender.wst.Append = true
log4j.appender.wst.MaxFileSize = 5000KB
log4j.appender.wst.MaxBackupIndex = 10
log4j.appender.wst.layout=org.apache.log4j.PatternLayout
log4j.appender.wst.layout.ConversionPattern=%m%n
#
log4j.appender.savelog=org.apache.log4j.RollingFileAppender
log4j.appender.savelog.File=../logs/saveall.rl
log4j.appender.savelog.Append = true
log4j.appender.savelog.MaxFileSize = 5000KB
log4j.appender.savelog.MaxBackupIndex = 10
log4j.appender.savelog.layout=org.apache.log4j.PatternLayout
log4j.appender.savelog.layout.ConversionPattern=%m%n
#
log4j.category.savelog.feb-webspective=DEBUG, wst
log4j.additivity.savelog.feb-webspective=false
log4j.category.savelog=INFO, savelog
log4j.additivity.savelog=false
#
```

2. Change the line from:

   ```
   log4j.rootCategory=INFO, A1
   ```

   to:

   ```
   log4j.rootCategory=INFO, A, savelog
   ```

   This adds the logging capability to the `savelog.feb-webspective` key.

7.5 Creating Events from Log Files

Use information contained in log files to generate events and alarms in Operations Center.

For example, assume a management tool monitors Web sites. This tool performs tests against Web sites and then stores the test result information in a file. In general, this type of file has a standard layout and contains information that can generate alarms or events in Operations Center.

To create events from log files:

1. Understand and parse the data.

   How is event data represented in the log file? If one row represents an entire event, parse the line and assign field names to that data. An event that is represented by multiple rows probably requires more sophisticated parsing.

2. Gather and send the data.

Many companies employ a utility called Stail, which monitors the end of log files and usually is written in Perl for portability. After adding new lines to the log file, the utiity reads, transposes, and sends the data to a server on a specific TCP/IP port. Using this utility correctly requires a clear understanding of the layout and meaning of every item in the log file. The data is read into memory and formatted to conform to a NOC Universal adapter data stream. The formatted data stream is sent to the NOC Universal adapters port.

The data stream consists of the required header, field/value pairs and the required footer. There is usually no mining capability so all events are sent with a class of Sync. Operations Center assigns the real class based on the `EventConsoleName` adapter properties.

7.6 Understanding Alarm Operations and Event Status

The Script adapter implements Acknowledge and Close alarm operations, but typically customers do not use them. Instead, they often create their own operations using scripting and custom operations. For details, see "Modifying Element and Alarm Menus" in the *Operations Center 5.5 Server Configuration Guide*.

When `originating_event_id` (see Section 7.3.2, "Data Fields," on page 214) and `originating_tec_hostname` data fields contain values for an event, Acknowledge and Close operations are available as right-click operations on the corresponding Operations Center alarm.

Acknowledge and Close operations affect the alarm by updating its status to *ACK* and *CLOSE*, and attempts to post back to the host server. The alarm no longer effect the element's condition, and alarm severity is unchanged.

To turn off the posting process to the host, set the adapter's *Post Status Changes to TEC* property to `false`. For property information, see Section A.32, "NetIQ Operations Center Universal," on page 339.

8 Establishing Console Connections

Console capability is available in Operations Center for some elements, depending on the adapter and element type.

If an undefined console type is requested by the Console option, the default console script definition is used to call the `telnet` class within Operations Center. This class utilizes two basic parameters: `host` and `port`. It also takes optional arguments: a name to be used as the title of the console window, and an icon to display. If these two optional arguments are not provided, then the console displays with the standard Operations Center icon and the title "Terminal."

The Operations Center console registry can be extended with script definitions for additional emulation modes for each type of console that a management system (or its adapter) can request.

The following sections provide instructions for using the Console capability:

8.1 Opening a Console Connection

When available and enabled, the *Console* option is available from the *Element* menu option, or by right-clicking an element. To view the list of adapters with console capability, see Section 8.2.2, "Assigning Emulation Modes to Adapters," on page 222.

The *Console* option runs a script from the Console registry that opens a console window to the application or management system. The selected script depends on the console type (also known as emulation mode). It is possible to configure the console type for some adapters. Other adapters have a required console type. For example, if an element's console emulation type is vt320, then the console script named vt320 runs.

Only a single console window for the element can exist at one time.

To open a console connection:

1. In the *Explorer* pane, navigate to any element for the management system adapter.
2. Right-click the element and select *Console*.

   The console option is available only for specific adapters that have enabled console capability

8.2 Setting Up Console Connections

To configure a console connection to a management system:

1. Add a Console Definition for each type of console that a management system (or its adapter) can request.

   For instructions, see Section 8.2.1, "Adding Scripts to the Console Registry," on page 220.

If the Operations Center console requests a console type that has not been defined in Console Definitions, then the default console script runs.

2 Hard code the console type in the management system or edit the adapter's hierarchy file.

For instructions, see Section 8.2.2, "Assigning Emulation Modes to Adapters," on page 222.

3 Perform any necessary modifications as required based on adapter property or element flag settings.

For information, see Section 8.2.3, "Adapter and Element Property Considerations," on page 225.

8.2.1 Adding Scripts to the Console Registry

Console Definitions are found under *Administration* > *Server* > *Console Definitions*. From there, they can be created, edited or deleted. You will want to add a script for each type of console that a management system (or its adapter) can request.

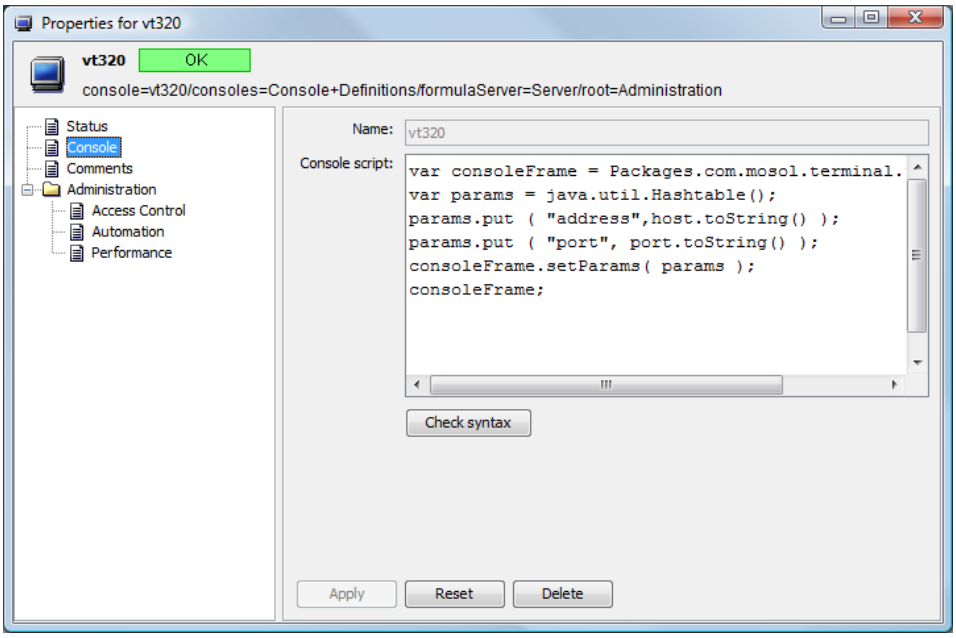
The default standard console script definition (named `default`) calls the `telnet` class within Operations Center:

```
var consoleFrame = formula.util.TelnetFrame();
 var params = java.util.Hashtable();
 params.put( "targetName", element.getName() );
 params.put( "address", host.toString() );
 params.put( "port", port.toString() );
 params.put( "targetIcon", element.getLabel().getIcon() );
 consoleFrame.setParams( params );
 consoleFrame;
```

The `telnet` class utilizes two basic parameters: `host` and `port`. It also takes optional arguments: a name to be used as the title of the console window, and an icon to display. If these two optional arguments are not provided, then the console displays with the standard Operations Center icon and the title "Terminal."

If the script evaluates to a `java.awt.Frame` object, then the Operations Center console tracks it in the log files.

It is advisable to review and if necessary, add Console script definitions, to ensure the Console feature works as expected with your management systems.



The console definition name matches the console emulation type. In the above figure, the definition name is vt320. If an element sets its console emulation type to vt320 when it wants to open a console, then the vt320 console script is used. This functionality is useful if you need to extend the default console script in some way, such as to resolve and address as in the case of GNAT or DHCP addressing or some other scheme.

As an example, the following code excerpt shows the script used for the Tndm6530 terminal emulation:

```
TeemWorldBean = Packages.TWBean.TeemWorldBean;
 var frame = java.awt.Frame( "Console for " + element.getName() );
 var twb = TeemWorldBean( frame );
 twb.setInitialEmulation( TeemWorldBean.TWB_TA6530 );
 twb.construct();
 twb.connect( host, port );
 frame;
```

The console type, also known as emulation mode, is configurable in some existing adapters and is hard coded in others. For more information, see Table 8-1.

To add a new console script definition:

1. In the *Explorer* pane, expand *Administration* > *Server*.
2. Right-click *Console Definitions*, then click *Create Console*.
3. Specify a name (case insensitive) that matches the emulation type in the *Name* field.

4 Enter the script content.

This script runs in the Operations Center console, passing context information that the console might require to display to users. The available context information is:

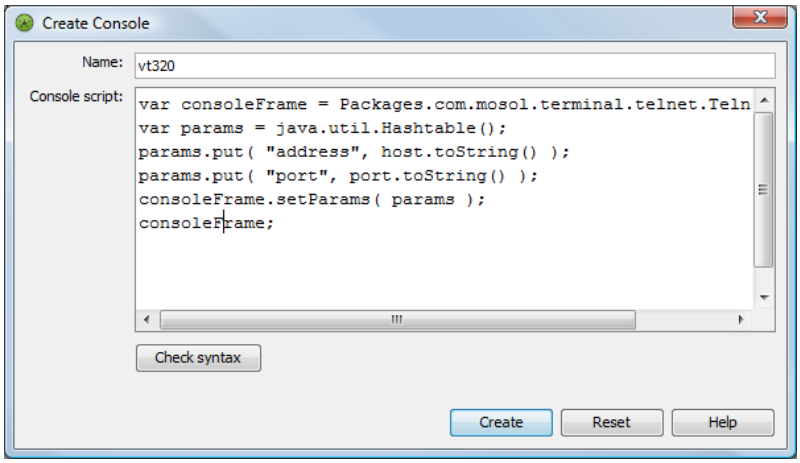
host: The hostname (or IP address) of the destination host.

port: The port number of the destination host.

element: (Optional) The element for which the console displays.

targetName: (Optional) A name to be used as the title of the console window.

targetIcon: (Optional) An icon to display.



5 Click *Check Syntax* to identify any errors.

6 Click *Create*.

8.2.2 Assigning Emulation Modes to Adapters

The emulation mode is configurable for some adapters and must be hard coded for others. For information about console capability and requirements, see Table 8-1 on page 223.

Configuring the type and port values requires setting parameters within the adapter's hierarchy file by using parameter tags.

The following code excerpt shows an example generator for the PATROL adapter that assigns the console type vt440:

```
<!-- Element breakout into host/appl/inst/parm  -->
  <generator affected="yes" class_from_field="hostType" field="host" hold="yes">
    <generator affected="yes" class="patrolAppl" field="appl" hold="yes">
      <generator affected="yes" class="patrolInst" field="inst">
        <generator affected="yes" class="patrolParm" field="parm"/>
      </generator>
    </generator>
    <param name="console" value="vt440"/>
    <param name="port" value="555"/>
  </generator>
</hierarchy>
```

To configure the emulation mode for an adapter:

1 To hard code an emulation type, define the `console` and `port` parameters in the management system.

The specific steps to defining these parameters depends on the specific management system.

2 To configure an adapter's emulation type:

2a Open the adapter's hierarchy file in a text editor.

The hierarchy file used by an adapter is specified in the Properties for an adapter. For more information about adapter properties, see Appendix A, "Adapter Property Reference," on page 279.

For information about hierarchy files, see Chapter 9, "Using the HierarchyFile," on page 227.

2b Add the required parameters to define the emulation mode.

For information about emulation parameters for each adapter type, see Table 8-1 on page 223.

Table 8-1 shows the current console capability and parameters used by the various adapters that exist within Operations Center, as well as additional information that can modify the console behavior.

Table 8-1 *Console Capability and Parameters for Adapters*

Adapter	Console Parameters
BMC Software PATROL Enterprise Manager (PEM)	**Console Capability:** Yes, only if the element has a defined host property. PEM can define a set of connection information for hosts/ports. It first checks if the management system determines the consoles that an element can connect to. **Console:** vt320, default for your telnet application, or set MODL parameter `console=`*`type`*. **Host:** Set parameter to `host=`*`hostname`*. **Port:** 23 or set MODL parameter to `port=`*`port_number`*.
BMC Software PATROL	**Console Capability:** Yes, only if an element is a Patrol host element or if it has a parent that is a Patrol host element. **Console:** vt320, default for your telnet application, or set MODL parameter in the adapter's hierarchy file to `console=`*`type`*. **Host:** Either this element or any of its parents. **Port:** 23 or set MODL parameter to `port=`*`port_number`*.
Computer Associates (CA) Spectrum	**Console Capability:** Yes, if the model element is a device and is not a network. **Console:** vt320, or the default for your telnet application **Host:** From the model's NET_ADDRESS attribute. **Port:** Hard coded to 23.
Computer Associates Unicenter	**Console Capability:** Yes, only if the element can be pinged or has an "address type" property. All console parameters are fully defined within the elements themselves.
Cisco Information Center (CIC)	**Console Capability:** Yes, only if the element has a defined host property. **Console:** vt320, default for your telnet application, or set MODL parameter to `console=`*`type`*. **Host:** Set parameter to `host=`*`hostname`*. **Port:** 23 or set MODL parameter to `port=`*`port_number`*.

Adapter	Console Parameters
HP OpenView Operations for UNIX	**Console Capability:** Yes for all node elements only. **Console:** vt320, or default for your telnet application. **Host:** Any ITO node itself. **Port:** Hard coded to 23.
HP OpenView Network Node Manager	**Console Capability:** All submap elements can console. Symbol elements can only console if the NetView management system has set the `canConsole` property for that element. **Console:** Hard coded to vt320; uses the default for your telnet application if not defined. **Host:** Gathered from element attributes. **Port:** Hard coded to 23.
IBM Micromuse Netcool	**Console Capability:** Yes, only if the element has a defined host property. **Console:** vt320, default for your telnet application, or set MODL parameter to `console=`*`type`*. **Host:** Set parameter `host=`*`hostname`*. **Port:** 23 or set MODL parameter to `port=`*`port_number`*.
IBM Tivoli NetView	**Console Capability:** All submap elements can console. Symbol elements can only console if the NetView management system has set the `canConsole` property for that element. **Console:** Hard coded to vt320, uses the default for your telnet application if not defined. **Host:** Gathered from element attributes. **Port:** Hard coded to 23.
IBM Tivoli T/EC	**Console Capability:** Yes, only if the element has a defined host property. **Console:** vt320, default for your telnet application, or set MODL parameter to `console=`*`type`*. **Host:** As defined in this element. **Port:** 23 or set MODL parameter to `port=`*`port_number`*.
IBM Tivoli T/EC+, Database Edition	**Console Capability:** Yes, only if the element has a defined host property. **Console:** vt320, default for your telnet application, or set MODL parameter to `console=`*`type`*. **Host:** As defined in this element. **Port:** 23 or set MODL parameter to `port=`*`port_number`*.
NetIQ AppManager	**Console Capability:** No.

Adapter	Console Parameters
NetIQ Operations Center Event Manager	**Console Capability:** Yes, only if an element is an Agent or if it has a parent that is an Agent. **Console:** The default for your telnet application, or set within the actual Agent as a parameter type from the Operations Center Event Manager (NOCEM) system. **Host:** Set within the NOCEM system for that Agent's host. **Port:** Set within NOCEM and based on if using TCPIP service or TELNET service.
NetIQ Operations Center InterConnection Adapter (ICA)	**Special Case:** The ICA adapter obtains the connection information of the remote system's viewable elements, so it is based on the adapters set up by those systems.
NetIQ Operations Center Universal Adapter	**Console Capability:** Yes, only if the element defines a host property. **Console:** vt320, default for your telnet application, or set MODL parameter to `console=`*`type`*. **Host:** An element's defined hostname, if one exists. **Port:** 23 or set MODL parameter to `port=`*`port_number`*.

8.2.3 Adapter and Element Property Considerations

To successfully establish a console connection with some management systems, there are adapter property or element flags that must be set correctly.

The following sections review required adapter and element property configurations necessary for console communications:

- "HostTokens Adapter Property" on page 226
- "Configuring Windows NT Servers" on page 226
- "Element Console Flags in the Management System" on page 226

HostTokens Adapter Property

Because some adapter properties can affect console connectivity, verify that all properties related to connectivity are specified correctly in adapter definitions.

For example, the BMC Software PATROL Enterprise Manager, Cisco Information Center, and Netcool adapters' `HostTokens` property is a list of PATROL Enterprise Manager token values that determine console connectivity. If Operations Center finds one of these tokens in an alarm, it uses that token to assign connectivity.

Configuring Windows NT Servers

The BMC Software PATROL Enterprise Manager adapter's ServicesFile property is relevant for administrators who want the console capability. The default, /etc/services, is appropriate for UNIX systems.

To configure Windows NT systems:

1. Copy the `/etc/services` file to a location on the NT server.
2. Change the value of the adapter's `ServicesFile` property to reference this file.

Element Console Flags in the Management System

Depending on the management system, specific element types ultimately control their own capabilities, which means that some elements for an adapter can perform the console operation, while others cannot. The console operation capability is controlled through the elements' capability flags or "CAPS" flags that are set in the management system. For consoles, this is CAP_CONSOLE. Elements that set this flag to `True` do not display a console operation if the element is in the busy state (`condition = USAGE_BUSY`).

As an example, the Tivoli NetView adapter has two element types: *Submap Elements* and *Symbol Elements*. Submap elements always set the console flag to True, but the Symbol elements first check to see if the Tivoli NetView management system has set an attribute for particular elements. This attribute, `canConsole`, must be `True` for the element to display the console operation in the Operations Center menus.

9 Using the HierarchyFile

Many network and system management systems maintain and update models of the physical elements that they manage, and Operations Center uses this information to create and display element hierarchies.

However, some management systems cannot identify the discrete element origins of the events that they generate, nor have the capacity to sort the events into a relationship structure that users can easily understand.

In cases such as this, Operations Center uses HierarchyFiles to interpret and organize the events reported by management systems that cannot identify discreet element origins. Without a HierarchyFile, Operations Center would represents everything reported by these management systems as a single element.

For example, hierarchy file configurations are used with adapters to:

- define a logical structure for grouping discovered objects found by using alarm fields
- filter events by status or severity
- define custom properties that are not included in the standard alarm and element property pages

The following sections describe how to use a hierarchy file to model objects and events:

9.1 Understanding Adapter HierarchyFiles

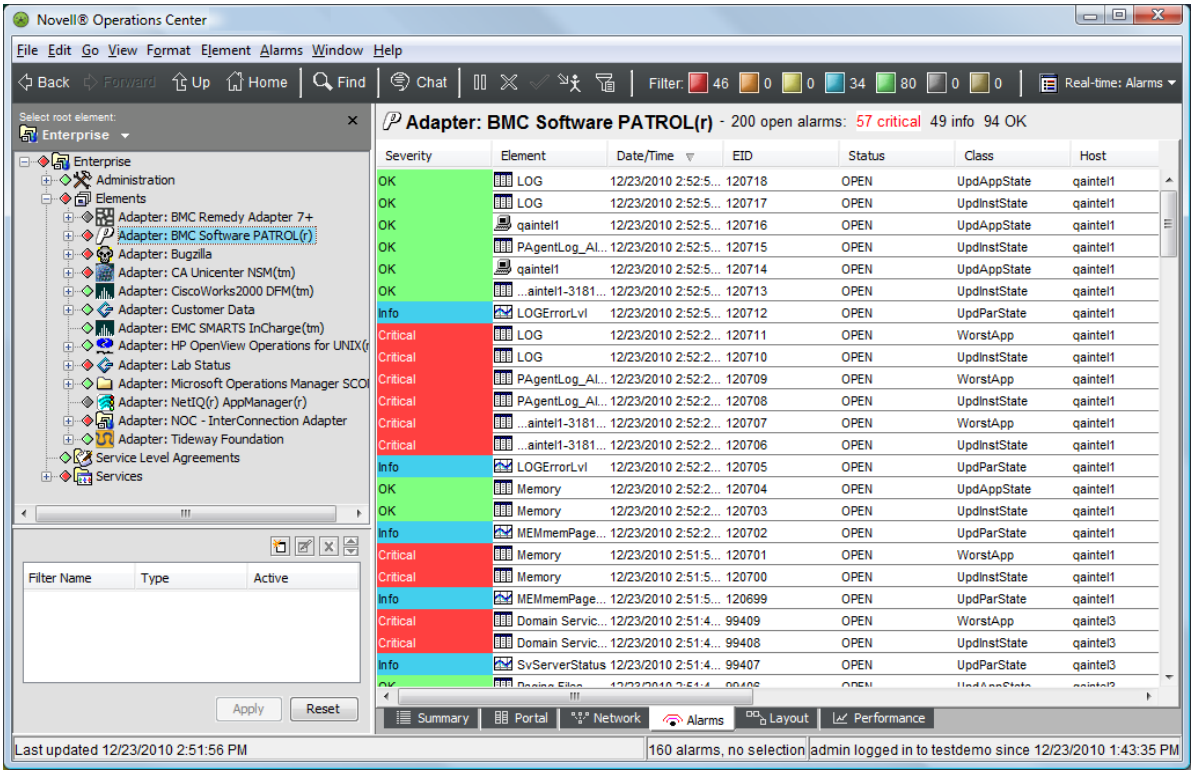
A HierarchyFile is an XML file that structures data streams received from various management systems and generates a hierarchy of elements which are used to create rational groupings of events and alarms.

When an Operations Center adapter receives a stream of events from a management system, these events are first translated into Operations Center alarms. Each alarm is processed using the adapter's HierarchyFile; the alarm cascades down through each rule, either matching or not matching criteria defined for elements. When matched, the alarm displays in Operations Center for the affected element.

Although every alarm displays at many levels in the hierarchy, each has only one affected element. For example, alarms display in the *Alarms* view for both the affected element and the element's parent. The affected element is the lowest level in the entire hierarchy that potentially can be affected by a specific alarm.

The name of the affected element displays in the *Element* column of the *Alarms* view, as shown in Figure 9-1. As alarms pass through the HierarchyFile, they attach to one or more elements and the affected attribute is set to yes. These elements each become a potential affected element. Operations Center then selects the lowest-level (most specific) of these potential elements as the affected element.

Figure 9-1 *Alarms View*



9.2 Modifying HierarchyFiles

For adapters that leverage a HierarchyFile, an example is provided in the /*OperationsCenter_install_path*/database/examples directory. These examples are automatically updated and overwritten when a new version of Operations Center or a patch release is installed, so it is important to save a copy in the */OperationsCenter_install_path*/database directory and make any changes there.

To modify an adapter HierarchyFile

1. Copy the adapter's example HierarchyFile from *OperationsCenter_install_path*/database/examples to the */OperationsCenter_install_path*/database directory.
2. Make the desired changes to the HierarchyFile.
3. Update the `Hierarchy File` property setting in the *Adapter Properties* for each adapter that uses the HierarchyFile.

9.3 Verifying Custom Property Values

To verify the custom property values:

1. Create an adapter and set its Hierarchyfile to the modified file (such as the example shown above).
2. Add the custom properties as display columns. For example, *Count* and *Counter*.
3. Start the adapter.
4. View the properties for the adapter element.
5. Open the custom property page.
6. Verify that the property values listed in the Custom property page match those defined in the Hierarchyfile.

9.4 HierarchyFile DTD Reference

Managed Objects Definition Language (MODL) is an XML-based markup language used to create HierarchyFiles for Operations Center adapters. MODL uses the XML model to assign meaning to events received from management systems. The HierarchyFile reflects both the nature of information received from a management system and the processing logic of the Operations Center system.

The MODL consists of a DTD (Data Type Definition) that follows rules for structuring an XML document.

TIP: To view the most recent DTD file, open the file */OperationsCenter_install_path*/database/examples/Hierarchy_2.0.DTD in a text editor.

The HierarchyFile DTD defines the rules, or grammar, for an XML document and it's tags. It explains how the HierarchyFile interacts with the adapter and the Operations Center software to create the end product: a representation of the event stream from the network management system displayed in the Operations Center console.

Table 9-1 provides an overview of the main HierarchyFile XML tags, that are used to provide a representation of the event stream from the network management system.

Table 9-1 *HierarchyFile XML Tags*

XML Tag	Description
<hierarchy>	The root XML tag. The HierarchyFile must contain a `<hierarchy>` XML tag that surrounds one or more `<group>`, `<filter>`, `<generator>`, or `<test>` XML tags. For more information about the hierarchy XML tag, see Section 9.4.1, "<hierarchy>," on page 231.
<group>	The `<group>` XML tag uses filters to create depth within the element hierarchy. Each group displays as an element in Operations Center. Nested generators and groups define a hierarchy of elements and alarms under the group's element. If there are no filters, all alarms attach to the group. Group tags can contain `<filter>`, `<fref>`, `<param>`, `<generator>`, or other `<groups>` tags. For more information about the group XML tag, see Section 9.4.2, "<group>," on page 231.
<generator>	The `<generator>` XML tag dynamically creates Operations Center elements as alarms arrive, based on specific alarm field values. A new element displays when an incoming alarm field value does not match any values of existing Operations Center elements. Generator tags can contain `<filter>`, `<fref>`, `<param>`, `<groups>`, or other `<generator>` tags. For more information about the generator XML tag, see Section 9.4.3, "The <generator> Tag," on page 232.
<filter>	The `<filter>` XML tag uses tests and fields to control which events apply to groups and generators.The filter is not a repository for an alarm; an alarm cannot attach to a filter. Filter tags can contain `<filter>`, `<fref>`, `<field>`, `<test>`, `<tref>`, or other `<filter>` tags. Filters must contain at least one child tag of any type. For more information about the filter XML tag, see Section 9.4.4, "filter," on page 234.

The following sections explain the HierarchyFile XML tags and attributes that constitute the rules used to structure and present information in a useful form:

- Section 9.4.12, “property,” on page 239
- Section 9.4.13, “value,” on page 241

9.4.1 <hierarchy>

The HierarchyFile must contain a <hierarchy> root XML tag that surrounds one or more groups, generators or filters. This XML tag contains the entire hierarchy intended for the information received from a management system.

The hierarchy declaration is:

```
<!ELEMENT hierarchy (group|filter|generator|test)+ >
<!ATTLIST hierarchy
   case (yes|no) "yes">
```

A <hierarchy> XML tag can have <group>, <filter>, <generator>, or <test> XML tags as children. The plus character (+) indicates that the incoming data must have a <hierarchy> XML tag containing one or more of the children XML tags, in the order listed. The <hierarchy> tag has no attributes.

An anonymous <filter> XML tag (one without a name attribute), defined immediately after the <hierarchy> XML tag, globally controls all events that any contained <group> or <generator> XML tags can process. If a candidate event does not match the criteria defined in the top-level filter, the event does not display in Operations Center.

For details on usage, see Section 9.4.4, “filter,” on page 234.

9.4.2 <group>

A <group> is a named tag that uses filters to create depth within the element hierarchy. Each group displays as an element in Operations Center. Groups can contain other groups, but the contained groups only receive the subset of alarms that passed their parent group’s filters. If there are no filters, all alarms attach to the group.

The group tag declaration is:

```
<!ELEMENT group (filter|fref|group|generator|param|properties|pref)* >
<!ATTLIST group
  name CDATA #REQUIRED
  class CDATA #REQUIRED
  affected (yes|no) "no"
  rollup CDATA #IMPLIED
  rollupParameters CDATA #IMPLIED>
```

Group tags can contain <filter>, <fref>, <generator>, <param>, <properties>, or other <groups> tags.

Table 9-2 describes <group> attributes.

Table 9-2 *group Attributes*

Attribute	Type	Description
name	Required	Assigns the group name, which displays in the element’s distinguished name (DName).

Attribute	Type	Description
class	Required	Assigns the group class, which displays in the element's distinguished name (DName). Assigning a class can be useful later when assigning an icon to the group, adding pop-up menu operations, and other class-sensitive features.
affected	Optional	Provides a way to declare that the affected element for this alarm potentially is this group, if a more detailed element does not claim the alarm instead.
rollup	Optional	Specifies a condition roll-up algorithm to call from the `algorithms.xml` file which replaces the default behavior of highest severity. When an algorithm is set using the roll-up attribute, it cannot be reset to the default algorithm through the Operations Center console, and can only be changed to another named algorithm.
rollupParameters	Optional	An additional parameter depending on the algorithm called by the roll-up attribute.

In the following example, the Consoles group applies a filter and a generator. The filter only accepts alarms whose affected element is console-capable. The generator creates a Operations Center element for each unique *ems* field in the incoming alarms.

```
<group name="Consoles" class="Consoles">
   <filter>
     <test type="element" expr="console" compare="true" />
   </filter>
   <generator field="ems" class_from_field="class" hold="yes" />
</group>
```

9.4.3 The <generator> Tag

The `<generator>` XML tag dynamically creates new Operations Center elements as alarms arrive, based on specific alarm field values.

For `<generator>` tags, the process of attaching an alarm creates a new Operations Center element only if the incoming field value is unique (does not match any existing element).

New elements created by a generator are not necessarily the affected element. Each alarm can have only one affected element. As an alarm moves through the hierarchy, many Operations Center elements might become potential affected elements. The affected element is the one to which the alarm finally attaches and the affected attribute equals yes.

The generator declaration is:

```
<!ELEMENT generator (filter|fref|generator|group|param|properties|pref)* >
<!ATTLiIST generator
  field CDATA #REQUIRED
  class CDATA #IMPLIED
  class_from_field CDATA #IMPLIED
  hold (yes|no) "no"
  affected (yes|no) "yes"
  case (yes|no) "yes"
  rollup CDATA #IMPLIED
  rollupParameters CDATA #IMPLIED>
```

Generator tags can contain `<filter>`, `<fref>`, `<group>`, `<param>`, `<properties>`, `<pref>`, or other `<generator>` XML tags.

Table 9-3 describes <generator> attributes.

Table 9-3 *generator Attributes*

Attribute	Type	Description
field	Required	The name used for the Operations Center element.
class or class_from_field	Required	Assign the group class name or use `class_from_field` to dynamically derive the element's class from the value of a specified alarm field.Set either the `class` or `class_from_field` attribute, but not both. In Operations Center, class determines the icon used to represent the element and others with the same class.
hold	Optional	The generated element can result in an unlimited number of generated elements over time. This can cause generated elements to "age out" after a period of time specified in the adapter settings. These elements can reappear if a new alarm requires it. Set `hold="yes"` to have the elements created by this generator remain in the Operations Center console until the adapter is turned off.
affected	Optional	Indicates the created element is potentially the affected element for an alarm if a more detailed (lower level) element does not claim the alarm instead.
case	Optional	The case attribute establishes the case sensitivity globally for the XML file. Set `case="yes"` if the field values are case-sensitive. Set `case="no"` if field values are not case-sensitive. If there is no case attribute for the <field> or <generator> attribute, the case attribute in the <hierarchy> XML tag applies. If there is no case attribute for the <hierarchy> XML tag, the default is Yes. `<!ELEMENT hierarchy (group|filter|generator|test)+ > <!ATTLIST hierarchy   case (yes|no) "yes">`
rollup	Optional	Specifies a condition roll-up algorithm to call from the `algorithms.xml` file which replaces the default behavior of highest severity. When an algorithm is set using the roll-up attribute, it cannot be reset to the default algorithm through the Operations Center console, and can only be changed to another named algorithm.

The following example creates a Operations Center element for every unique value in the `group` alarm field. New elements have a class equal to the value of the groups field.

```
<generator field="group" class_from_field="group" case="yes|no" hold="yes"
rollup="average" rollupParameters="average" />
```

<filter> and <fref>) definitions used in generator statements perform exactly as they do in a <group> statement, controlling the events that pass down a branch of the hierarchy.

The following example shows nested generator statements. Each <generator> statement creates child elements for the elements created by the parent <generator> statement.

```
<generator field="class" class_from_field="class" hold="yes">
    <generator field="ems" class_from_field="class" hold="yes" />
    <group name="servers" class="server" />
    <group name="hosts" class="host" />
</generator>
```

Operations Center selects as the actual affected element the lowest-level and most specific of all potential affected elements. In the following example of nested <generator> statements, alarms attach to only those elements created by the inner statement, <generator field="sub_source">.

```
<generator field="source" class="source" affected="yes">
    <generator field="sub_source" class="sub_source" affected="yes"/>
</generator>
```

9.4.4 filter

A <filter> controls which alarms can attach to group or generator elements. The filter is not a repository for an alarm; an alarm cannot attach to a filter. Specify one or more filters. A filter can test an alarm's fields, or it can test other information known when new alarms arrive.

The filter tag declaration is:

```
<!ELEMENT filter (filter|fref|field|test|tref)+ >
<!ATTLIST filter
name ID #IMPLIED
operator (and|or) "and">
invert (true|false) "false">
```

Filter tags can contain <filter>, <fref>, <field>, <test>, <tref>, or other <filter> tags. Use nested filters or <fref> tags for more conditional filtering. Filters must contain at least one child tag of any type.

Table 9-4 describes <filter> attributes.

Table 9-4 *filter Attributes*

Attribute	Type	Description
name	Optional	Assigns a name to the filter. Name a filter to reference it later without having to reenter the entire filter statement, by using the <fref> tag. For more information about the fref XML tag, see Section 9.4.7, "fref and tref," on page 237.
operator	Required	Assigns Boolean logic to two filter conditions. The operator can be either and or or. The default is and. An incoming alarm must pass all filters when using the and operator. An incoming alarm must pass at least one filter when using the or operator.
invert	Optional	Invert the result of the comparison by setting the attribute to false. The default is true, which leaves the comparison unchanged.

In the following example, an incoming alarm must satisfy only one of the two conditions named within, as the operator is set to or. An incoming alarm with a class equal to MEGAT or ORN200 attaches to the group, which is the parent element of the filter. Use filters in a similar way with <generator> tags.

```
<filter name="nps" operator="or">
    <field name="class" operator="equals" value="MEGAT" />
    <field name="class" operator="equals" value="ORN200" />
</filter>
```

Since filters can contain other filters or <fref> filter references, it is possible to build complex, multipart tests for inclusion or exclusion. They can act as gatekeepers at the top of different branches of a hierarchy. In complex HierarchyFiles, cap long branches of the hierarchy with filters to control the alarms pass on to their child elements. The appropriate parts of the hierarchy parse alarms, as shown in the example in Section 9.4.2, "<group>," on page 231.

9.4.5 field

A <field> XML tag tests specific fields in an incoming alarm against a target value. A field cannot contain any other XML tags.

The field declaration is:

```
<!ELEMENT field EMPTY>
<!ATTLIST field
  name CDATA #REQUIRED
  operator (equals|less|less_or_equals|greater|greater_or_equals|
    contains|starts_with|ends_with) "equals"
  value CDATA #REQUIRED
  case (yes|no) "yes"
  invert (true|false) "false">

<filter>
```

Table 9-5 describes <field> attributes.

Table 9-5 *field Attributes*

Attribute	Type	Description
name	Required	Specifies a field to test in the incoming alarm.
operator	Required	Compares the value of the selected alarm field to the value attribute using one of the following operators: `(equals|less|less_or_equals|greater|greater_or_equals|contains|starts_with|ends_with)`
value	Required	Assigns Boolean logic to two filter conditions. The operator can be either `and` or `or`. The default is `and`. An incoming alarm must pass all filters when using the `and` operator. An incoming alarm must pass at least one filter when using the `or` operator.
case	Optional	The case attribute establishes the case sensitivity globally for the XML file. Set `case="yes"` if the field values are case-sensitive. Set `case="no"` if field values are not case-sensitive. If there is no case attribute for the <field> or <generator> tag, the case attribute for the <hierarchy> XML tag applies. If there is no case attribute for the <hierarchy> XML tag, the default is `Yes`. `<!ELEMENT hierarchy (group|filter|generator|test)+ >` `<!ATTLIST hierarchy` `case (yes|no) "yes">`
invert	Optional	Invert the result of the comparison by setting the attribute to `false`. The default is `true`, which leaves the comparison unchanged.

In the following example, the filter uses the field tag to check the *Severity* field of incoming alarms for a severity value that is less than 3:

```
<filter name="ccfilter">
    <field name="severity" operator="less_than" value="3" />
</filter>
```

The following example shows how nested filters can be used and is a snippet from a HierarchyFile.

```
<filter name="sch" operator="and">
   <filter operator="or">
      <field name="text" operator="contains" value="FT002204" />
   </filter>
   <filter operator="and">
      <field compare="false" name="rule" value="Citicorp" />
      <field compare="false" name="text" operator="contains" value="IM" />
   </filter>
</filter>
```

9.4.6 test

The <test> XML tag checks the properties of an alarm's affected element. A test can determine whether an element has a specific capacity, such as console, or it can run a script that returns a Boolean result.

The test declaration is:

```
<!ELEMENT test EMPTY>
<!ATTLIST test
  name ID #IMPLIED
  type (element|script) #REQUIRED
  expr CDATA #REQUIRED
  invert (true|false) "false">
```

Table 9-6 describes the <test> attributes.

Table 9-6 *test Attributes*

Attribute	Type	Description
name	Optional	Assigns a name to the test. Name a test to reference it later without having to reenter the entire filter statement, by using the <tref> tag. For more information about the tref XML tag, see Section 9.4.7, "fref and tref," on page 237.
type	Required	Defines whether the test runs directly on the element or if a script runs.
expr	Required	The name of the property to test or script load command.
invert	Optional	Invert the result of the comparison by setting the attribute to `false`. The default is `true`, which leaves the comparison unchanged.

In the following example, a filter uses the <test> tag to check if an alarm's affected element has the console capability. The type attribute is set to `element` and the `expr` attribute identifies the value console. The filter selects all affected elements that have console capability.

```
<filter>
  <test type="element" expr="console" compare="true" />
</filter>
```

Set the compare attribute to `false` to return all affected elements that do not have console capability.

9.4.7 fref and tref

<fref> and <tref> XML tags are references that call previously defined filters or tests. This avoids the inconvenience of redefining the same filter or test more than once in the body of the XML.

The fref declaration to call a named filter is:

```
<!ELEMENT fref EMPTY>
<!ATTLIST fref
  name IDREF #REQUIRED>
```

The tref declaration to call a named test is:

```
<!ELEMENT tref EMPTY>
<!ATTLIST tref
  name IDREF #REQUIRED>
```

Table 9-7 describes <fref> and <tref> attributes.

Table 9-7 *fref and tref Attributes*

Attribute	Type	Description
name	Required	The name of the test or filter to call.

In the following example, the <fref> calls the filter named `ccfilter`, as previously defined in the example shown in Section 9.4.4, "filter," on page 234.

```
<fref name=ccfilter />
```

9.4.8 pref

A <pref> is a properties reference. It references a <properties> XML tag by name which contains a definition of one or more element properties. Referencing a <properties> XML tag with <pref> effectively reuses the <properties> definition at the point of the <pref>.

```
<!ELEMENT pref EMPTY >
<!ATTLIST pref
    name CDATA #IMPLIED >
```

9.4.9 page

A <page> allows grouping the <property> and <value> XML tags that it contains for a specific named page displayed in the browser. If there is no name, the default page applies.

```
<!ELEMENT page (property|value)+ >
<!ATTLIST page
    name CDATA #IMPLIED >
```

9.4.10 param

A <param> XML tag is an arbitrary parameter which can be assigned to elements. Use params to support adapter-specific behavior. Specify a <param> tag within a <group> or <generator> tag.

The param declaration is:

```
<!ELEMENT param EMPTY>
<!ATTLIST param
  name CDATA #REQUIRED
  value CDATA #REQUIRED>
```

Table 9-7 describes <param> attributes.

Table 9-8 *param Attributes*

Attribute	Type	Description
name	Required	The name of the parameter.
value	Required	The value of the parameter.

For example, an administrator can change the information displayed for elements in the Operations Center console's *Notes* column. The default information displayed in the *Notes* column is a summary of active alarm counts for a specific element or a text message if there is only one alarm associated with the element.

In the following example, the *Notes* field for each generated element displays the text message of the most recently received or changed alarms:

```
<generator field="hostname">
  <param name="lastMessageAsNotes" value="true" />
</generator>
```

For information about available parameters, see Section 9.5, "Parameter Reference," on page 241.

For examples using parameters, see Section 9.8, "Example: Mining Performance Data," on page 245 and Section 9.9, "Example: SCM Matching," on page 246.

9.4.11 properties

<properties> is a container for a group of page tags. It can be optionally name, which allows a <pref> tag to reference the named <properties> tag and effectively reuse its definition.

The properties declaration is:

```
<!ELEMENT properties (page)+ >
<!ATTLIST properties
   name CDATA #IMPLIED
```

Table 9-9 *properties Attributes*

Attribute	Type	Description
name	Optional	Assigns a name that allows a <pref> XML tag to reference the named <properties> XML tag and effectively reuse its definition.

9.4.12 property

A <property> defines an element property with a value that is computed from an alarm field for a group or generated element.

The property declaration is:

```
<!ELEMENT property EMPTY >
<!ATTLIST property
    name CDATA #REQUIRED
    field CDATA #REQUIRED
    function
(LAST|FIRST|AVG|SUM|LOWEST|HIGHEST|MIN|MAX|ROLLINGAVG|PSEUDOAVG|LASTWITHVALUE)
#REQUIRED
    parameter CDATA #IMPLIED
    computeOnGet (true|false) "false"
    computeOnNoAlarms CDATA #IMPLIED
    computeLastDateTimeStampField CDATA #IMPLIED
>
```

Table 9-10 *property Attributes*

Attribute	Type	Description
name	Required	Name of the property.
field	Required	Alarm field used to compute the value.
computeOnGet	Required	Defines when to compute the custom property. If `false` (the default), computing the property value occurs at alarm creation, update and deletion time only for the elements to which the alarm is attached. Although the Operations Center console *Alarms* view displays alarms for all child elements as well, the computation does not include child element alarms. If `true` computes the property value when the value is needed, such as for a property page or a `getAttr()` call. The computation is for all alarms that can display for an element, including those attached to child elements.
computeOnNoAlarms	Optional	
computeLastDateTimeStampField	Optional	Evaluates date and time stamp values in the specified alarm field to determine which alarm should be considered last, compared to other alarms already received. Use with the LAST function to determine which alarm is the most recent.

Attribute	Type	Description
function	Required	Type of computation to be performed on the alarm field. Specify one of the following function types: ◆ **LAST:** Field value in the most recent alarm activity.* ◆ **LASTWITHVALUE:** Field value in the most recent alarm activity* where the field exists. ◆ **FIRST:** Field value in the oldest alarm. ◆ **AVG:** Field values averaged over all existing alarms. Returns a numeric value. ◆ **ROLLINGAVG:** Field value averaged over the last *n* (where *n* is given in the parameter attribute) alarm creates and changes (but not deletes) against the element. *n* defaults to 10.Returns a numeric value. ◆ **PSEUDOAVG:** Alarm field value pseudo-averaged over the last *n* (where *n* is defined in the parameter attribute) alarm creates and changes (but not deletes) against the element. *n* defaults to 10. Returns a numeric value. ◆ **SUM:** Alarm field value summed over all existing alarms for the element. ◆ **LOWEST:** Minimum field value in alphanumeric sort order across all alarm activity.* Returns a string. ◆ **HIGHEST:** Maximum field value in alphanumeric sort order across all alarm activity.* Returns a string. ◆ **MIN:** Minimum field value in numeric order across all alarm activity. Returns a numeric value. ◆ **MAX:** Maximum field value in numeric order across all alarm activity. Returns a numeric value. *Alarm activity refers to all alarm creations, changes or deletions for the element. Where indicated, some functions interpret the value of the alarm field as string data or as numeric data. If alarm field is a string, but the function uses a numeric conversion, it will try to convert it to a numeric value.

Use the following requirements and best practices when defining properties:

- Property names must be unique across custom property pages.
- Do not use the same names as a Operations Center properties (such as Element, Condition, and so on).
- Do not specify multiple properties computing the same function against the same alarm field.
- Any alarm field used to compute a property is a mandatory alarm field.

The following code computes custom property values from the alarm field after alarm creation, change, or deletion:

```
<properties>
   <page>
      <property field="q1" function="SUM" name="q1:SUM" computeOnGet="true" />
      <property field="q1" function="AVG" name="q1:AVG" computeOnGet="false" />
      <property field="q1" function="MIN" name="q1:MIN" computeOnGet="true" />
      <property field="q1" function="MAX" name="q1:MAX" computeOnGet="false" />
    </page>
</properties>
```

The following statement looks for the most recent value of the *OSISeverity* alarm field, but uses a comparison on the *creationTimestamp* field to determine what alarm should be considered the last. If the date value inside the *creationTimestamp* field is not greater than the previously received alarm, then the alarm is skipped and does not determine the value of the *LastSeverity* property.

```
<property field="OSISeverity" function="LAST"
computeLastDateTimeStampField="creationTimestamp" name="LastSeverity"/>
```

For an example using properties, see Section 9.7, "Example: Custom Properties from Alarm Fields," on page 244.

9.4.13 value

A `<value>` defines an element property created with a literal value for a group or generated element. The declaration is:

```
<!ELEMENT value EMPTY >
<!ATTLIST value
    name CDATA #REQUIRED
    value CDATA #REQUIRED >
```

Table 9-11 *value Attributes*

Attribute	Type	Description
name	Required	Name of the value.
field	Required	Literal value for a group or generated element.

9.5 Parameter Reference

9.5.1 Alarm Summary Parameters

In the Operations Center console, alarm summaries display in the *View* pane title bar for all views except *Alarms*, using the format: 200 open alarms, 50 critical, 100 major, 50 minor. When an element contains only one alarm, the summary displays the actual alarm.

Use the `alwaysShowAlarmCounts` parameter to use the alarm summary format for single alarms instead of displaying the actual alarm.

Table 9-13 describes `<param>` tags for alarm summary counts.

Table 9-12 *param tags for Performance Data*

param Name	Type	Description
alwaysShowAlarmCounts	Required	If `true` or `yes`, calculates summary information for an alarm. Note that alwaysShowAlarmCounts overrides `<param name="notesAsLastMessage">`.

To view summary information for a single alarm in the format: *1 open alarm, 1 critical*, use the following <param> tag:

```
<param name="alwaysShowAlarmCounts" value="true"/>
```

9.5.2 Performance Data Parameters

Elements created from <generator> tag can have performance data attached to them using a few <param> tags. The <param> tags define how to read the performance data from a database. After the performance data is attached to the element, performance charts can be created using the usual methods.

The following requirements must exist to populate the performance chart with database information:

- Performance data exists in a relational database which can be accessed from the Operations Center server.
- The database has a JDBC driver which is listed the `formula.db.jdbc.drivers` property within the `/OperationsCenter_install_path/config/Formula.properties` file.
- IData is returned from a SQL query with a time stamp in the first column and the remaining columns numeric. The data must be in ascending order by time stamp.

Table 9-13 describes <param> tags for performance data.

Table 9-13 *param tags for Performance Data*

param Name	Type	Description
Performance.Label	Required	The series name of the performance data.
Performance.Metrics	Required	A semi-colon delimited list of database column names. These are the names of the data that appears in Operations Center and must match a column names in the query.
Performance.Query	Required	The SQL query retrieving the desired data with a time stamp in the first column and integers or floats in remaining columns. With the exception of the time stamp column, the alias of each data column must match a name in the Performance.Metrics list or it will not be chartable. Data must be returned in ascending order by time stamp.
Performance.Database	Optional	Full URL of database to query, including the machine name, port and database name. This defaults to the setting of `formula.db.jdbc.url` in the `Formula.properties` file.
Performance.User	Optional	The user name for the database login. This defaults to the setting of `formula.db.user` in `Formula.properties`.
Performance.Password	Optional	The password for the database user account. This defaults to the setting of `formula.db.password` in the `Formula.properties` file.

Dynamic values from the alarms and generated can be inserted into param values to customize the series name, column names and SQL query for each element and alarm. These dynamic values are:

- **{element.name}** Name of element being generated.

- **{query.from}, {query.to}** Starting and ending time stamps specified by user when creating performance chart. These should be formatted in the query to correspond to the time stamp format used by the given database.
- **{alarm.field}** The value of any defined field of the alarm generating the element. For example, {alarm.source} would be the source of the alarm while {alarm.hostname} would be the hostname of the machine that raised the alarm. The names of fields available for an alarm depend on the adapter.

Use the following requirements and best practices when using params for performance data:

- Each param value must be on a single line with no line breaks.
- All single quotes must be doubled in the SQL query.
- Do not use greater than and less than symbols. Instead use `>` and `<`.
- Time stamps must be formatted using the syntax above, but might need a different format string depending on the database type. The formatting syntax is documented in the `java.text` package. See the online Java documentation at http://java.sun.com/j2se/1.3/docs/api/index.html (http://java.sun.com/j2se/1.3/docs/api/index.html).

9.5.3 SCM Matching Parameters

The MODL file can generate matches to target elements using structure or source values in the Service Configuration Manager (SCM).

Table 9-14 describes `<param>` tags for SCM data.

Table 9-14 *Parameters Used to Mine Values from the SCM*

Param Name	Description
match.*n*.type	Type of match to perform. Set to on of the following: `FIXED`, `REGEXP`, `LDAP`, `CLASS`
match.*n*.expr	Used to create the variable portion of a match expression or static DName using template expansion. If no source elements are provided, this expression uses the element generated as the element for template expansion.
match.*n*.root	If no source elements are provided, the root element is used for generating the location of expression root in the resulting matcher, or for the fixed portion of the static match generated. As an example, if the root is specified for a REGEXP matcher, the resulting final match expression concatenates the expression value with the root value.

In the following example, a *Switch* element is created with the class of *telcom_switch*. Then, using param declarations, child elements are matched when an element is found that has a grandparent with gen-device class and

```
<group name="Switch" class="telcom_switch">
    <param name="match.0.type" value="FIXED" />
    <param name="match.0.expr"
value="gen_device=${formula.util.encodeURL($parent.parent.name)}" />
    <param name="match.0.root" value="gen_container=Switches/demo=SNMP+Manager/
root=Elements" />
</group>
```

For a more extensive example using SCM matching parameters, see Section 9.9, “Example: SCM Matching,” on page 246.

9.6 Example: Defining a Dynamic Element Structure

When writing a HierarchyFile, consider the degree to which the current environment is expected to change. When expecting minimal change, use the <group> XML tag to explicitly name elements from incoming alarms.

Assume that the network has only one host. This single group handles assigning host names to incoming alarms.

```
group description="Hosts" class="Hosts">
   <group description="hostname" class="hostname"
      <filter>
         <field name="hostname" value="BSCMserver"/>
      </filter>
   </group>
</group>
```

If you expect that the element structure will grow or change, or the exact nature of incoming alarms from a management system is unknown, use the <generator> tag.

The following example creates an element container (parent element) for each host name. The Hosts group contains a single <generator> statement which creates new elements when an incoming alarm has a new value for the *hostname* field. Each new element name will contain the *hostname* field value.

```
<group description="Hosts" class="Hosts">
   <generator field="hostname" class="hostname"/>
</group>
```

The above example avoids having to constantly update the HierarchyFile with specific hostnames. The generator dynamically creates a hostname element whenever it detects a new host and automatically adds it to the hierarchy, as well as automatically assign incoming alarms to the new element.

Alternatively, use the class_from_field attribute in place of the class attribute to name the element's class using field values.

<generator> statements can create more complicated structures, either by nesting or in combination with the other tags such as <groups> and <filters>.

9.7 Example: Custom Properties from Alarm Fields

The following HierarchyFile example uses a group statement to first create a parent element named *Source* and populates it with alarms having a *count* alarm field with a value of greater than 0. Then it creates a *Custom Properties* property page with a list of properties whose values are computed using the *Count* and *Counter* alarm fields.

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE hierarchy PUBLIC "-//NetIQ, Inc.//DTD hierarchy 2.0//EN" "http://
www.ManagedObjects.com/dtds/hierarchy_2.0.dtd">

<hierarchy case="yes">
    <group name="Source" class="Source" affected="yes">
        <filter operator="and" invert="false">
            <field name="count" operator="less_or_equals" value="0" case="no"
invert="false" />
        </filter>
        <properties name="Custom Property Pages">
           <page name="Custom Properties">
              <property name="countHIGHEST" field="count" function="HIGHEST"
computeGetOn="false" />
```

```
              <property name="countLOWEST" field="count" function="LOWEST"
computeGetOn="false" />
              <property name="countFIRST" field="count" function="FIRST"
computeGetOn="false" />
              <property name="countLAST" field="count" function="LAST"
computeGetOn="false" />
              <property name="countSUM" field="count" function="SUM"
computeGetOn="false" />
              <property name="countAVG" field="count" function="AVG"
computeGetOn="false" />
              <property name="countMIN" field="count" function="MIN"
computeGetOn="false" />
              <property name="countMAX" field="count" function="MAX"
computeGetOn="false" />
              <property name="counterHIGHEST" field="counter" function="HIGHEST"
computeGetOn="false" />
              <property name="counterLOWEST" field="counter" function="LOWEST"
computeGetOn="false" />
              <property name="counterFIRST" field="counter" function="FIRST"
computeGetOn="false" />
              <property name="counterLAST" field="counter" function="LAST"
computeGetOn="false" />
              <property name="counterSUM" field="counter" function="SUM"
computeGetOn="false" />
              <property name="counterAVG" field="counter" function="AVG"
computeGetOn="false" />
              <property name="counterMIN" field="counter" function="MIN"
computeGetOn="false" />
              <property name="counterMAX" field="counter" function="MAX"
computeGetOn="false" />
            </page>
          </properties>
      </group>
  </hierarchy>
```

9.8 Example: Mining Performance Data

The following HierarchyFile example uses a `generator` statement to create a new element for each unique value in the `hostname` alarm field, then queries the *samples* database to mine the average response time from performance data.

```
<generator field="hostname">
  <param name="Performance.Label" value="Average Performance for {element.name}" />
  <param name="Performance.Metrics" value="Avg. Response Time (ms);Avg. Number of
Transactions to {element.name}" />
  <param name="Performance.Database" value="jdbc:db2:ewm" />
  <param name="Performance.User" value="admin" />
  <param name="Performance.Password" value="sesame" />
  <param name="Performance.Query" value="select time,value*1000 ''Avg. Response
Time (ms)'',trans ''Avg. Number of Transactions to {element.name}'' from samples
where host=''{alarm.hostname}'' and metric=''{alarm.source}'' and time
&gt;= ''{query.from,date,yyyy-MM-dd-HH.mm.ss.000000}'' and time
&lt;= ''{query.to,date,yyyy-MM-dd-HH.mm.ss.000000}'' order by time" />
</generator>
```

The above example equates to the following SQL query with line breaks added:

```
select time, value*1000 ''Avg. Response Time (ms)'', trans ''Avg. Number of
Transactions to {element.name}''

from samples

where
```

```
  host=''{alarm.hostname}'' and metric=''{alarm.source}'' and time &gt;=
''{query.from,date,yyyy-MM-dd-HH.mm.ss.000000}'' and time &lt;=
''{query.to,date,yyyy-MM-dd-HH.mm.ss.000000}'' order by time
```

9.9 Example: SCM Matching

The following HierarchyFile example uses nested `generator` statements to create an element hierarchy 4 levels deep from 4 generations of parent class names. Under the hierarchy built from class names, are two main group elements: Networking (with Switch and Router child elements) and Performance.

- *structure.parent.parent.parent.name*
  - *structure.parent.parent.name*
    - *structure.parent.name*
      - *structure.name*
        - Networking
          - Switch
          - Router
        - Performance

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE hierarchy PUBLIC "-//NetIQ, Inc.//DTD hierarchy 2.0//EN" "http://
www.ManagedObjects.com/dtds/hierarchy_2.0.dtd">
<hierarchy case="yes">

    <generator affected="no"
class_from_field="structure.parent.parent.parent.objectClass"
field="structure.parent.parent.parent.name">

        <generator affected="no"
class_from_field="structure.parent.parent.objectClass"
field="structure.parent.parent.name">

            <generator affected="no"
class_from_field="structure.parent.objectClass" field="structure.parent.name">

                <generator affected="yes" class_from_field="structure.objectClass"
field="structure.name">

                    <group name="Networking" class="net_lan" >
                        <group name="Switch" class="telcom_switch">
                            <param name="match.0.type" value="FIXED" />
                            <param name="match.0.expr"
value="gen_device=${formula.util.encodeURL($parent.parent.name)}" />
                            <param name="match.0.root"
value="gen_container=Switches/demo=SNMP+Manager/root=Elements" />
                        </group>
                        <group name="Router" class="device_router">
                            <generator field="#set($s=${structure.id}%32)$s"
class="gen_port">
                                <param name="match.0.type" value="FIXED" />
                                <param name="match.0.expr" value="port=${name}/
server=${formula.util.encodeURL($parent.parent.parent.name)}" />
                                <param name="match.0.root"
value="gen_container=Routers/demo=SNMP+Manager/root=Elements" />
                                <param name="script"
value="element.DisplaySourceElements=true" />
                            </generator>
```

```
                        </group>
                    </group>

                    <group name="Performance" class="gen_container" >
                        <param name="match.0.type" value="REGEXP" />
                        <param name="match.0.expr" value="[^\x2f]*/
${parent.objectClass}=${formula.util.encodeURL(${formula.util.escapeRegExp($parent
.name)})}" />
                        <param name="match.0.root" value="Hosts=Hosts/
tec=Tivoli+T%2FEC/root=Elements" />
                        <param name="match.1.type" value="LDAP" />
                        <param name="match.1.expr" value="(cn=${parent.name})" />
                        <param name="match.1.root" value="gen_container=Hosts/
script=Sitescope/root=Elements" />
                    </group>
                    <group name="Tickets" class="gen_action" >
                        <param name="match.0.type" value="CLASS" />
                        <param name="match.0.expr"
value="(&amp;(cn=${parent.name})(objectClass=*))" />
                        <param name="match.0.root" value="remedy=ARS+Help+Desk/
root=Elements" />
                    </group>

                </generator>

            </generator>

        </generator>

    </generator>

</hierarchy>
```

10 ORB Installation

Integrating Operations Center software with some management systems requires the Operations Center ORB software. ORB software is installed during the installation or upgrade of the Operations Center software using the single Operations Center Installation CD.

Operations Center software uses ORB software to act as a broker between a client request for a service from a distributed object or component and the completion of that request. Components can find out about each other and exchange interface information while they are running. The ORB software is a required interface between adapters and several management systems to enable the proper interchange of information.

10.1 About ORBs

Operations Center requires ORBs to integrate with the following management systems:

Table 10-1 *ORB Installation Sections*

Management System	See For Installation
CA Unicenter	Section 10.2, “UniORB for CA Unicenter,” on page 252
HP OpenView Network Node Manager	Section 10.3, “OvORB for HP OpenView Network Node Manager,” on page 263
HP OpenView Operations for UNIX (OVO)	Section 10.4, “OVOORB for HP OpenView Operations for UNIX,” on page 265
IBM Tivoli Enterprise Console (T/EC)	Section 10.5, “TecORB for IBM Tivoli Enterprise Console,” on page 267
IBM Tivoli NetView	Section 10.6, “NvORB for IBM Tivoli NetView,” on page 274

The Operations Center installer provides options for installing ORBs. The exceptions are NvORB and UniORB, which must be installed manually. No additional procedures are required, unless noted. The following sections provide information about each ORB including system requirements and troubleshooting information.

After installing the ORB software, the next step is defining adapters, which provide the interface between Operations Center software and managed elements. See

- Section 10.1.1, "Troubleshooting ORBs: Identifying Port Conflicts," on page 250
- Section 10.1.2, "Using ORB Log Files," on page 250
- Section 10.1.3, "Using Multi-Homed Servers," on page 251

10.1.1 Troubleshooting ORBs: Identifying Port Conflicts

The ORB installation program configures a default TCP/IP port number on the server. If another application on the server uses the default port, then change the port number used by the ORB.

Table 10-2 summarizes the default ports used by each ORB and provides links to the relevant ORB section.

Table 10-2 *Default Ports*

ORB	Default Port
NvORB for IBM Tivoli NetView (page 274)	1572
OvORB for HP OpenView Network Node Manager (page 263)	1572
OVOORB for HP OpenView Operations for UNIX (page 265)	1578
TecORB for IBM Tivoli Enterprise Console (page 267)	1576
UniORB for CA Unicenter (page 252)	1580

Before installing any ORB, use the `netstat` command to determine if the default port is already in use. If the default port is already in use, then change the port number used by Operations Center software. See the relevant ORB section to more information about the `netstat` command for each ORB.

10.1.2 Using ORB Log Files

Log files are available to review ORB activity and history. The following log file commands are available for NvORB and OvORB:

Table 10-3 *Log File Arguments*

Argument	Argument Description
–LogLevel level	Sets the type of entries logged to the file. Logging includes all entries for that level and any levels below it: 0 = Error 1 = Warning 2 = Info 3 = Verbose 4 = Exhaustive
–LogFile file name	Changes the default log file name to the specified file name.
–LogAppend	Appends additional entries to the current log file instead of creating a new one.

10.1.3 Using Multi-Homed Servers

Some Operations Center software installed on a multi-homed server might require configuration changes to ensure that components on different machines communicate correctly. A multi-homed server is a server with more than one IP address.

Understanding Multi-Homed Servers

On multi-homed servers, one IP address is the primary or default IP address, and the other IP addresses are virtual or nonprimary IP addresses. Operations Center software installed on a system with multiple IP addresses can use the primary IP address to communicate with a server located on another host.

However, additional configurations for some ORBs are necessary to ensure using one of the multi-homed server's nonprimary (virtual) IP addresses. Otherwise, communication errors can result from a component publishing an inappropriate IP address or hostname to another component.

IMPORTANT: Management systems that integrate with Operations Center without using ORB software do not require any special configuration for multi-homed servers.

If all components communicate with one other using the primary IP address of the server on which they are running, no configuration changes are required. However, if one or more components communicate using virtual or nonprimary IP addresses, configuration changes are required.

For example, assume that a server recognizes an ORB located on a multi-homed server by one of its virtual IP addresses. By default, the ORB publishes references back to Operations Center software using the primary IP address of its server. In this situation, the server most likely cannot reach the primary IP address, resulting in communications failures.

Perform one of the following actions to resolve multi-homed server communication errors:

- Use the server's primary IP address
- Change the ORB settings to always publish references with a reachable hostname that the server can correctly resolve

Configuring the ORB for a Multi-Homed Server

ORBs for some management systems require additional configuration for multi-homed servers. Table 10-4 lists the management systems that require additional configuration and describes the modifications required for the ORB software:

Table 10-4 *ORBs Requiring Configuration for Multi-Homed Servers*

Management System	ORB	Configurations Required
Computer Associates (CA) Unicenter™	UniORB	Unicenter integrates with Operations Center software through the use of the UniORB. However, the UniORB does not support the ability to set the hostname using a start command. If the UniORB is located on a multi-homed server, Operations Center software must communicate with the ORB by means of that server's primary IP address.
IBM Tivoli NetView®	TecORB	See steps below for changing ORB settings.
HP OpenView™ Network Node Manager (NNM)	OvORB	See steps below for changing ORB settings.

To update ORB settings for affected management systems, append the following argument to the command that invokes the ORB:

start_command `-OAhost` *hostname*

where *start_command* is the start command for the associated management system ORB (for more information, see the corresponding ORB section in this guide), and *hostname* is the valid hostname that the server can use to communicate with the ORB.

For `-OAhost`, the capital letter O follows the hyphen, not a zero.

The Operations Center console automatically updates its published IP address to one that is acceptable to the server, so no configuration changes are required.

10.2 UniORB for CA Unicenter

UniORB is a Operations Center software component that wraps the Computer Associates Unicenter Enterprise Management System with a CORBA object interface, enabling access to many of Unicenter's core functions and data over a network.

The CA Unicenter adapter in Operations Center software connects to the UniORB over the network (or locally, if Unicenter and Operations Center software are installed on the same machine) to provide real-time access to Unicenter managed objects and events from any Web browser, using the Operations Center console. It is also possible to use Operations Center software to automate and contain Unicenter managed objects in one centralized location.

For instructions on creating and configuring CA Unicenter adapters after the UniORB is installed, see Section 3.10, “Computer Associates Unicenter,” on page 67.

10.2.1 System Requirements

Table 10-5 outlines the system requirements for installing the UniORB:

Table 10-5 *System Requirements*

OS	Description
Windows	◆ CA Unicenter 3.0 or 3.1 for Windows 2000 Unicenter installations for UNIX are not fully supported at this time. ◆ Windows 2000 ◆ Approximately 7MB available disk space

10.2.2 General Steps for Installation

This section provides a high-level overview of the installation procedure. Detailed installation procedures are provided in the remaining sections of this section. Click the hypertext links to directly access the relevant section.

To install UniORB:

1. Verify that you have the most current versions of the server and UniORB.
2. Uninstall previous versions of UniORB from the Unicenter server.
   For instructions, see Section 10.2.3, “Uninstalling a Prior UniORB Installation,” on page 254.
3. Verify the default port is available.
   For instructions, see Section 10.2.4, “Verifying that the Default Port is Available,” on page 254.
4. (Optional) Implement a modified copy of Unicenter `common.cnf`.
5. Install UniORB and the UniORB service using the Operations Center Installation CD.
   For instructions, see Section 10.2.5, “Installing UniORB,” on page 254.
6. Configure the UniORB service.
   For instructions, see Section 10.2.7, “Configuring the UniORB Service,” on page 257.

7 Start UniORB.

   For instructions, see Section 10.2.13, “Starting and Stopping the UniORB,” on page 262.

8 Define the Unicenter adapter.

   For more information about defining and using the Unicenter adapter, see Section 3.10, “Computer Associates Unicenter,” on page 67.

10.2.3 Uninstalling a Prior UniORB Installation

If a prior version of the UniORB is installed, use the following steps to uninstall it completely. This is a precautionary step to ensure that errors resulting from a prior installation do not occur during the new UniORB installation.

To uninstall a prior version of UniORB:

1 To remove registry entries that are currently set, from the desktop, click *Start* > *Programs* > *Managed Objects* > *UniORB* > *Deinstall Service*.

2 Double-click *Control Panel*, then click *Add/Remove Programs* and uninstall the UniORB.

10.2.4 Verifying that the Default Port is Available

The default port for UniORB is 1580. Prior to installing UniORB, verify that port 1580 is available.

To verify that port 1580 is available:

1 At a command prompt, enter:

```
netstat -an | find "LISTEN" | find "1580"
```

2 If the above command results in a line ending with the word LISTEN, then select a different port.

10.2.5 Installing UniORB

Install UniORB from the `/ORBS` directory on the Operations Center CD.

To install UniORB:

1 On the Operations Center CD, navigate to the `/ORBS/Unicenter/NT` directory.

2 Launch setup.exe.

10.2.6 Installation Considerations

Although the Installation Wizard does streamline the UniORB installation process, there are some options to consider when installing for the first time or when it is necessary to first uninstall an existing UniORB.

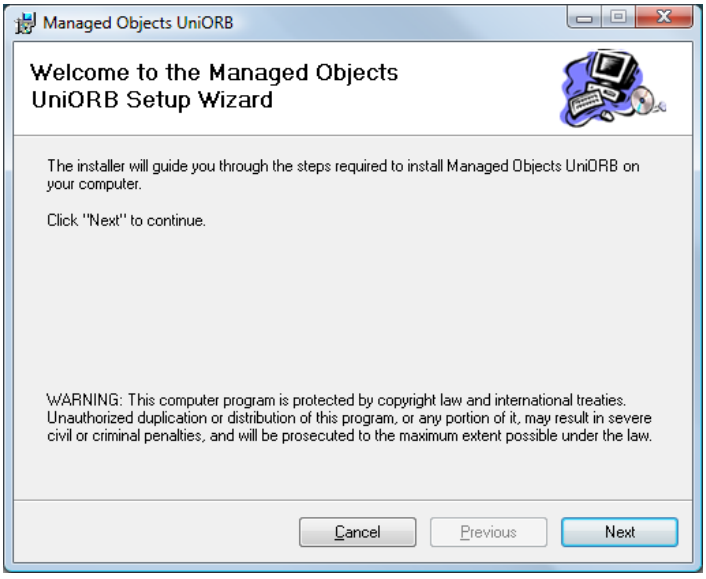
If the UniORB is already installed on the system, you must uninstall it before installing a new UniORB. If the Install Wizard finds an existing UniORB, the following screen displays:

Figure 10-1 *Operations Center UniORB Installation Wizard – Removing previously installed UniORB*

To remove the old UniORB, click *Remove Operations Center UniORB*, then click *Finish*. Afterwards, follow the steps again to launch the installer.

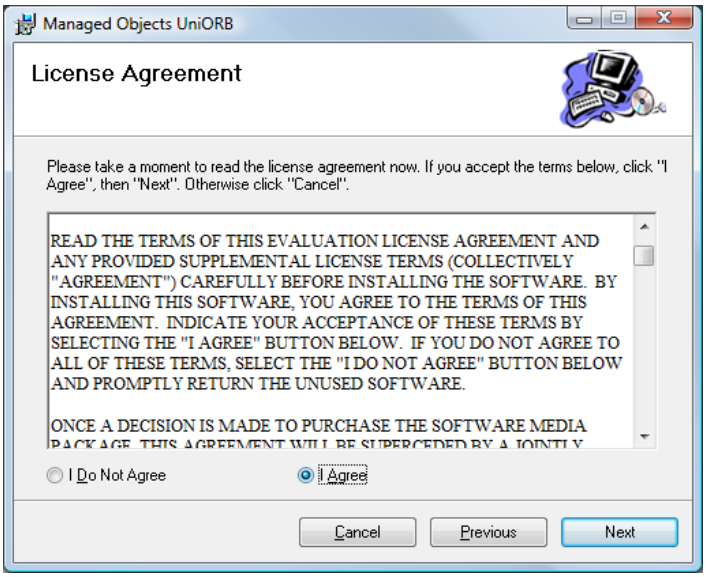
When UniORB is installed, either for the first time or after the removal process on an existing version is complete, the following screen displays:

Figure 10-2 *Operations Center UniORB Installation Wizard*



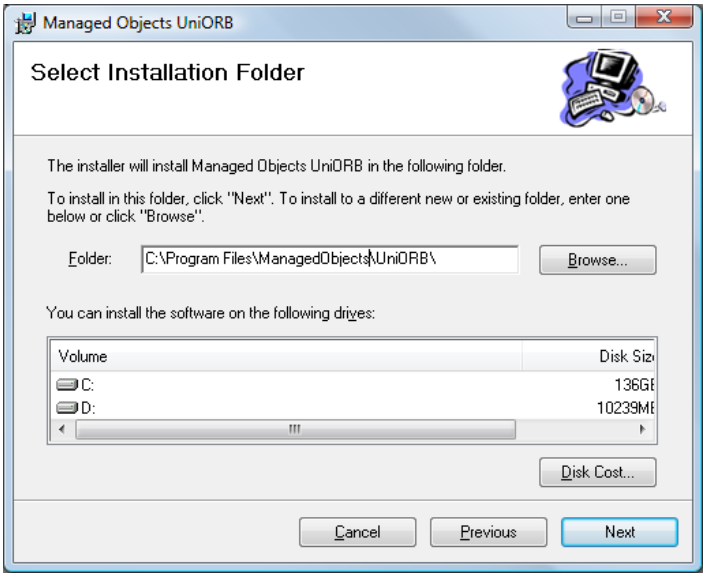
After you click *Next*, the following screen displays:

Figure 10-3 *Operations Center UniORB Installation Wizard – Agree to License Agreement*



Select *I Agree* to accept the license agreement., then click *Next*. The following screen displays:

Figure 10-4 *Operations Center UniORB Installation Wizard – Selecting the Installation folder*



Specify the installation folder and follow the prompts to complete the installation process.

10.2.7 Configuring the UniORB Service

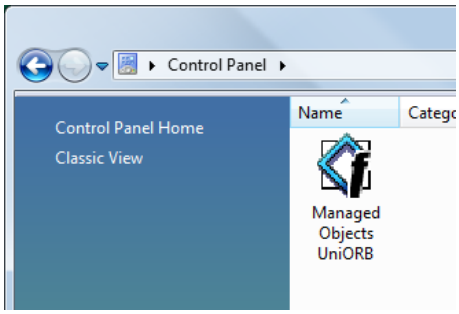
During the UniORB installation, a dialog box offers you the option of customizing the UniORB parameters. If this dialog box does not open, or if you want to manually reconfigure the UniORB, use the procedures in the following sections:

- "Manually Configuring the UniORB" on page 257
- "Setting the Startup Options" on page 259

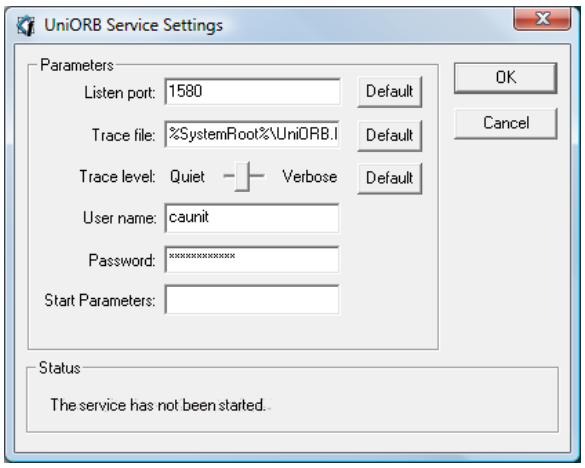
Manually Configuring the UniORB

To configure the UniORB:

1 From the desktop, click *Start* > *Settings* > *Control Panel* to open the Control Panel dialog box.



2 Double-click *Managed Objects UniORB* to open the UniORB Service Settings dialog box:



3 Specify the port (Listen port) on which to listen for network requests, the path to the `UniORB.log` (trace) file and the level of detail (trace level) in the UniORB log file.

If the TCP/IP port number 1580 is already in use on the Unicenter server, select a different port as described in Section 10.2.4, "Verifying that the Default Port is Available," on page 254.

4 In the *Trace File* field, specify the path to store the `UniORB.log` trace file.

   The default location is `c:/windows/UniORB.log`.

   If the Operations Center server is installed on the same machine as the UniORB, specify the `/OperationsCenter_install_path/logs` directory. This makes it easier to locate the log file later.

   If the Operations Center server is not installed on the same machine, specify the `/Program Files/OperationsCenter_install_path/UniORB` directory (the directory where the UniORB is installed).

5 Adjust the Trace Level slider used to represent the trace level of the `UniORB.log` file.

   When set completely to the right, the DEBUG level produces considerable trace data.

   For best general results, set the slider to position 4 or 5 (there are a total of six positions, 1 being the quietest).

6 Specify the start parameters, which are listed in Section 10.2.14, "Running UniORB from the Command Line," on page 262.

7 In the *User Name* and *Password* fields, specify the ID and password of a user who has access to the network shares that enable the UniORB to retrieve remote Unicenter Enterprise Manager events.

   This user can be an account in the local domain or a full domain account, but the user ID must have the authority to access all UNISHARE$ netnames on the machines where the Unicenter Enterprise Managers are installed.

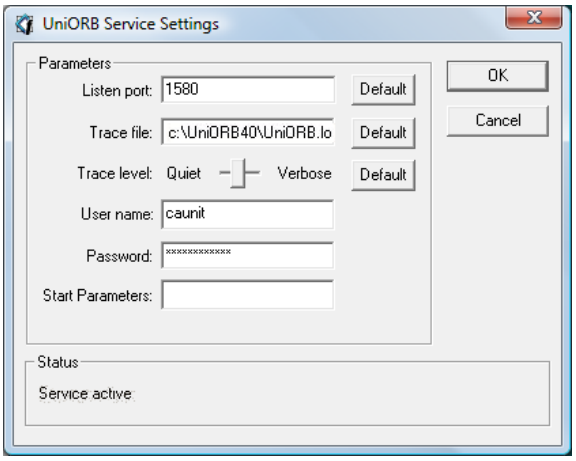
   The UniORB uses the supplied information to perform a dynamic login after startup. After the login, the UniORB maps a Universal Naming Convention (UNC) share name to the directory on each Unicenter server where Unicenter logs are kept, so the user ID must have the necessary authorization to map that UNC name. To a Unicenter administrator, these servers are Enterprise Management servers or EM servers.

   The user ID and password is not validated until the UniORB starts. Be careful when typing the password, as there is no password verification, and the password stored in the registry is masked. Also see Section 10.2.11, "Additional Issues Regarding User IDs and Passwords," on page 261.

8 Click *OK* to close the UniORB Service Settings dialog box.

Figure 10-2 shows a UniORB configuration with the required information. If the dialog box on the screen does not match what is shown in Figure 10-2, the most recent version of the UniORB is not installed.

Figure 10-5 *UniORB Service Settings*



Note the information displayed in the Status area at the bottom of the dialog box. The message Service active indicates that the service was configured through the Control Panel applet, installed through the *Start* menu option, and that the UniORB service has started.

Other messages that can display in the Status area are:

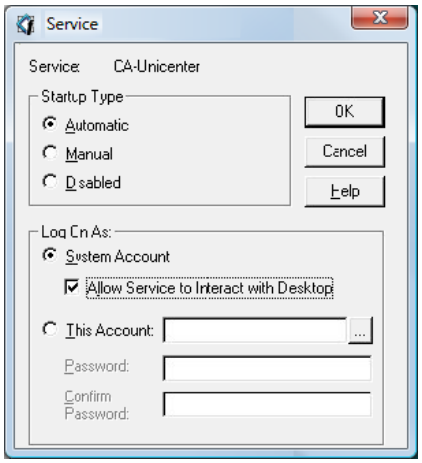
```
Service has not been installed.
```

```
Service is not active.
```

Setting the Startup Options

To set the startup options on UniORB:

1. From the desktop, click *Start* > *Settings* > *Control Panel* to open the Control Panel dialog box.
2. Double-click *Services* to open the Services dialog box.
3. Select *Managed Objects UniORB*, then click *Startup* to open the Service dialog box.



4. Select one of the following radio buttons:

   Automatic: Starts UniORB automatically on system startup.

   Manual: Start UniORB manually.
5. Verify that Log On As is set to System Account and that *Allow Service to Interact With Desktop* is selected.

   This log on setting is different from the user ID specified in the UniORB configuration. Do not change it to any other value.
6. Click *OK* to close the CA Unicenter Service dialog box.
7. Click *Close* to close the Services dialog box.

The correct configuration of the UniORB service, when viewed from the Control Panel Services applet, is to Log on as Local System Account, with *Interact With Desktop* selected.

If this configuration is not used, the UniORB does not see event messages from Unicenter, and state (color) changes are not propagated from Unicenter to Operations Center software. State information and additions and deletions display in Operations Center software when the adapter stops and restarts, but additional changes do not display. Operations Center software also does not recognize the dynamic creation and deletion of objects.

The UniORB service uses the System Account to log on and start several processes which together support Operations Center software's communication with Unicenter. Only one of these processes uses the real user ID and password supplied in the UniORB Control Panel applet.

10.2.8 Changing Registry Entries

Registry entries used by the UniORB service are set when the service is installed. Additional values are set when the Control Panel applet executes. If you are concerned that the registry values are not set correctly, simply perform again the steps listed in the previous section. It is not necessary to uninstall and reinstall the UniORB software to set these registry entries.

The registry entries used by the UniORB are located at: HKEY_LOCAL_Machine/SYSTEM/CurrentControlSet/Services/Operations Center UniORB

10.2.9 Repository Object Filtering

The UniORB class filtering feature provides the ability to subscribe to certain classes in Unicenter. Instead of bringing in all objects and hierarchies to Operations Center, you can exclude the object classes that are not of interest to you.

The `UniORB` command takes an optional parameter, `-f=`*`filename`*, where *filename* is a text file that lists TNG WorldView classes to omit. Optionally, identify superclasses by adding this suffix to the class name, such as `::ALLCHILDCLASSES`. Another option is to place comments in the file using the `#` character.

The following code is an example filter file:

```
# Example Uniorb Repository Class Filter Definition File
# (c) NetIQ, Inc. 2014
#
# For use with UniORB 3.51 and above only
#
# To define the repository filter file to the uniorb the -f parameter
# must be used in the uniorb start-up parameters, configurable via the
# Control Panel applet. There is no restriction on the filename or
# location of the filter file.
#
#
# File Contents
# -------------
# Any line prefixed with the # character is considered a comment and ignored
# Class names TO BE FILTERED OUT are listed, one per line
# Optionally the suffix ::ALLCHILDCLASSES can be used to indicate that the
# specified class and any subclasses should be filtered out.
#
# For example, to filter out the Agent class and any sub-classes;
# Agent::ALLCHILDCLASSES
#
Workstation
WBEM
IPSubnet
Ping
```

This repository filter file is separate (for backwards compatibility) from the event log filter file.

10.2.10 Automatic Service Dependencies

There might be situations in which the UniORB should depend on specific services such as Unicenter or SQL Server. If either service is shut down, the UniORB should be shut down. Additionally, the UniORB should restart after Unicenter or SQL Server has restarted.

To accomplish this dynamic service dependency, the UniORB checks a Registry leaf (DependOnServices), which is a comma-separated list of service names on which the UniORB service is dependant. When UniORB starts up, the value is compared with the current service configuration and all changes are applied automatically to the service definition.

10.2.11 Additional Issues Regarding User IDs and Passwords

The user ID is stored exactly as it is entered. Therefore, it is possible to specify an NT domain account as well as a local account. If you are concerned that the user ID is not being handled properly, enter the user ID in the form:

```
NT_domain \user_ID
```

where *NT_domain* is the domain name of the Domain Controller that authenticates the logon, and *user_ID* is a user name that must also have certain advanced user rights that are common to programs that execute as a service.

In general, members of the Administrators group have most of the rights needed. Those shown in Table 10-6 are commonly necessary. Those shown in bold print have been observed to be missing in some cases:

Table 10-6 *Rights Needed for Each Operating Platform*

Operating Platform	Advanced User Rights Needed to:
Windows NT Server	◆ Access this computer from the network ◆ Log on as a service ◆ Log on locally ◆ Act as part of the operating system
Windows 2000 Server	◆ Act as part of the operating system ◆ Log on as a service ◆ Log on locally ◆ Take ownership of files or other objects

In a few observed cases (on Windows 2000), it also might be necessary to ensure that the account used has the required privileges in the local domain, even if the account used is a domain account rather than a local domain account.

10.2.12 Configuration Notes

The UNC name mapped by the UniORB is the value defined by Unicenter during its installation. This UNC name must be the default value of UNISHARE$ for correct mapping to occur. This UNC name is mapped to the Unicenter root directory on the Unicenter server. Typically, this is: `C:/NSM`.

Scanning the log files in the `/NSM/LOGS` directory provides the information displayed by Operations Center software in the *Alarms* view. If the UniORB cannot map to the directory where the logs are located, Operations Center software does not display any alarms for Unicenter. In this situation, a message appears at the beginning of the UniORB log, possibly indicating a login failure, as well as a brief indication of the likely cause.

10.2.13 Starting and Stopping the UniORB

To start the UniORB:

1. From the desktop, click *Start* > *Settings* > *Control Panel* to open the Control Panel dialog box.
2. Double-click *Administrative Tools* to open the Administrative Tools dialog box.
3. Double-click *Services* to open the Services dialog box.
4. Right-click *Managed Objects*, click *UniORB*, then click *Start*.

   The service starts.

   To manually stop the UniORB, use the `mosstop` command.

10.2.14 Running UniORB from the Command Line

To view the command line syntax supported by the UniORB, issue the `uniorb` command with the -h parameter, as shown (the output of the command also displays):

```
C:> uniorb -h

Usage:
uniorb [options]

Options:
-h, --help            Show this message.
-i, --install         Install the service.
-u, --uninstall       Uninstall the service.
-c, --console         Run in console mode.
```

Use the –c parameter to invoke the UniORB from the command line for diagnostic purposes.

When the UniORB runs from the command line, two additional command line parameters might be useful for diagnostics. These options have the same functionality within all ORBs.

Table 10-7 *Command Line Parameters for Diagnostics*

Parameter	Meaning
–OAhost hostname	A valid hostname that the server can use to communicate with the ORB. A capital letter O, not a zero, follows the hyphen (–). This name instructs the ORB to always publish its references with the proper hostname, one that the server can correctly resolve and reach over the network.
–OAport port_number	A valid port number that the server can use to communicate with the ORB. Note that a capital letter O, not a zero, follows the –.

10.3 OvORB for HP OpenView Network Node Manager

OvORB is a Operations Center software component that wraps the HP OpenView Network Node Manager (NNM) network management system with a JAVA RMI object interface, thus enabling access to many of OpenView's alarm-oriented core functions and data over a network.

For instructions on creating and configuring HP OpenView Network Node Manager adapters after the OvORB is installed, see Section 3.12, "HP OpenView Network Node Manager," on page 89.

10.3.1 Preinstallation Notes

The HP OpenView NNM adapter in Operations Center software connects to the OvORB over the network or locally, if OpenView and Operations Center software are installed on the same machine. The OvORB provides real-time access to OpenView alarms from any Web browser using the Operations Center console. It is also possible to place OpenView alarms in a centralized location using the Operations Center software automation functions.

Accessing OpenView alarm information in Operations Center software requires running both the OvORB and the HP OpenView NNM adapter.

10.3.2 System Requirements

Table 10-8 outlines the system requirements for installing the OvORB:

Table 10-8 *System Requirements*

OS	Description
Windows	HP OpenView NNM 6.4, 7.0x, or 7.5 for Windows 2000
HP-UX (PA-RISC Only), Linux, and Solaris	HP OpenView NNM 6.4, 7.0*x*, or 7.5 For Linux only, RedHat Linux Application Server 2.1 (AS2.1) - 7.01 of NNM

10.3.3 Verifying that the Default Port is Available

The default port for OvORB is 1572. Prior to installing OvORB, determine if port 1572 is already in use on the OpenView server.

To verify port 1572 is available:

1. Enter one of the following at a command prompt:

   Windows: `netstat -an | find "LISTEN" | find "1572"`

   UNIX: `netstat -an | grep "LISTEN" | grep "1572"`

2. If either command results in a line ending with `LISTENING`, the port is in use; therefore, select a different port.

   For UNIX, select an available port number above 1000 if running the OpenView client as a user other than root.

10.3.4 Configuring HP OpenView NNM with OvORB

The ORB requires some manual configuration after the installer is finished. Note the file permissions must be at least `r-xr-xr-x`.

To configure the OvORB:

1. Register the OvORB as an OV managed process.

   From the command line, use the HP `ovaddobj` command to add the OvORB as an OV managed process. For example, if the OvORB is installed in the `D:/root/ovorb` directory, execute the command:

   ```
   ovaddobj D:/root/ovorb/lrf/ovorb.lrf
   ```

2. Copy the Operations Center Application Registration file to the OpenView `Registration` directory.

   For example, if the OvORB is installed in the `D:/root/ovorb` directory for Windows, execute the command:

   ```
   copy "D:/root/ovorb/registration/C/*" "C:/Program Files/HP OpenView/
   registration/C"
   ```

   For UNIX, execute the command is:

   ```
   cp /root/ovorb/registration/C/* /etc/opt/OV/share/registration/C
   ```

3. Issue the following command to start the OvORB:

   ```
   ovorb start
   ```

   The registration is a one-time process. When you register the OvORB as an OV managed process, it automatically stops and starts when OV is stopped and started.

4. Operations Center is now ready to connect the NNM adapter.

10.3.5 Loading Multiple OvORB Instances on a UNIX Server

Multiple OvORBs can run on the same UNIX server.

To install multiple OvORB instances:

1. Install the OvORB as described in the previous section.
2. When the OvORB Customizer opens, click the *Program Instance* drop-down list and select an instance of the OvORB installation.
3. Select from 1–10 based on the number of OvORBs previously installed.

   For example, if this is the first installation of an OvORB, the Program Instance is 1. For the second installation, the Program Instance is 2.

10.3.6 Starting and Stopping the OvORB

The OvORB is an OpenView managed process that starts and stops automatically. It is no longer started by the NNM as in previous adapter versions.

If the OvORB is not set up as an OV managed process, or if there is a need to restart the ORB, start or stop it via the command line.

To manually start the OvORB, enter the following HP NNM command:

```
/opt/OV/bin/ovstart ovorb
```

To manually stop the OvORB, enter the following HP NNM command:

```
/opt/OV/bin/ovstop ovorb
```

10.4 OVOORB for HP OpenView Operations for UNIX

OVOORB is an optional component of Managed Object's integration with HP OpenView Operations for UNIX (OVO™). When OVOORB is available on an OVO management server, it permits additional capabilities that are not possible without its presence.

The OVOORB enables bidirectional access to the OVO management server, including the ability to modify event status, such as event acknowledgement, ownership, changing event workflow and triggering automation.

OVOORB is implemented in Java, so it can install and run on any operating system supported by OVO.

- Section 10.4.1, "System Requirements," on page 266
- Section 10.4.2, "Verifying that the Default Port is Available," on page 266
- Section 10.4.3, "Configuring OVOORB," on page 266
- Section 10.4.4, "Starting and Stopping OVOORB," on page 266

For instructions on creating and configuring HP OpenView Operations for UNIX adapters after the OVOORB is installed, see Section 3.13, "HP OpenView Operations for UNIX," on page 99.

10.4.1 System Requirements

Table 10-9 outlines the system requirements for installing the OVOORB:

Table 10-9 *System Requirements*

OS	Description
HP-UX (PA-RISC Only)	◆ HP OpenView Operations 7.*x*, 8.*x* ◆ HP-UX 11.i As the ORB is Java-based, the HP box needs to comply with HP's requirements for java (kernel parameters and patches). For more information, see http://www.hp.com/products1/unix/java/. HP also provides a HP tool (HPjconfig) that can check and advise on compliance. For more information and downloads, see http://www.hp.com/products1/unix/java/java2/hpjconfig/index.html. ◆ Approximately 400MB available disk space
Solaris	◆ HP OpenView Operations 7.*x*, 8.*x* ◆ Solaris 2.9 Additional required patches for Solaris as indicated by Sun. ◆ Approximately 400MB available disk space

10.4.2 Verifying that the Default Port is Available

The default port for OVOORB is 1578. Prior to installing OVOORB, determine if port 1578 is already in use on the OVO server.

To verify port 1578 is available on Windows:

1 At a command prompt, enter:

```
netstat -an | grep "LISTEN" | grep "1578"
```

2 If the above command results in a line ending with the word LISTEN, then select a different port.

10.4.3 Configuring OVOORB

To configure the OVOORB:

1 If the TCP/IP port number 1578 is already in use on the OVO server, select a different port when prompted during ORB installation.

10.4.4 Starting and Stopping OVOORB

To start or stop the OVOORB:

1 To start the OVOORB, from the */OperationsCenter_install_path*/bin directory, enter:

```
ovoorb start
```

2 To stop the OVOORB, from the */OperationsCenter_install_path*/bin directory, enter:

```
ovoorb stop
```

10.5 TecORB for IBM Tivoli Enterprise Console

TecORB is an optional Operations Center software component integration with IBM Tivoli Enterprise Console (T/EC). When TecORB is available on a T/EC event server, it permits the following additional capabilities:

- Faster closure and acknowledgement of events from Operations Center software to T/EC, using `wsetemsg`
- The ability, through NOC Script, to populate Operations Center software with existing events from T/EC's event repository, using `wtdumper`

TecORB is implemented in Java, so it can install and run on any operating system supported by T/EC. It provides standard OMG CORBA 2.0 integration with many of T/EC's commands and functions without using the Tivoli framework.

- Section 10.5.1, "System Requirements," on page 267
- Section 10.5.2, "T/EC Rule Base Customization Considerations," on page 268
- Section 10.5.3, "Customizing T/EC for Integration with Operations Center software," on page 268
- Section 10.5.4, "Verifying that the Default Port is Available," on page 272
- Section 10.5.5, "Configuring TecORB," on page 273
- Section 10.5.6, "Restricting Access by IP Address," on page 273
- Section 10.5.7, "Starting and Stopping TecORB," on page 273

For instructions on creating and configuring IBM Tivoli Enterprise Console adapters after the TecORB is installed, see Section 3.15, "IBM Tivoli Enterprise Console (T/EC)," on page 108.

10.5.1 System Requirements

Table 10-10 outlines the system requirements for installing the TecORB:

Table 10-10 *System Requirements*

OS	Description
AIX	◆ IBM Tivoli Enterprise Console (T/EC) version 3.9 for IBM AIX ◆ IBM AIX 5.2 or 5.3 ◆ Approximately 30MB available disk space on the `/opt` file system
HP-UX (PA-RISC Only)	◆ IBM Tivoli Enterprise Console (T/EC) version 3.9 for HP-UX ◆ HP-UX 11i (11.1x) ◆ Approximately 30MB available disk space on the `/opt` file system
Linux	◆ RedHat Linux Application Server 2.1 (AS2.1)

OS	Description
Solaris	◆ IBM Tivoli Enterprise Console (T/EC) version 3.9 for Solaris ◆ Solaris 2.9 Additional required patches for Solaris. ◆ Approximately 30MB available disk space on the `/opt` file system

10.5.2 T/EC Rule Base Customization Considerations

This section provides an example rule that forwards and synchronizes all T/EC events with a server. It is a modification of the basic set of rules recommended by IBM Tivoli for synchronizing two T/EC servers.

Multiple TecORBs reporting to a Operations Center TEC adapter need separate event listening ports defined for each adapter. Also, each port must have a matching forwarding port in each tec_forward.conf in a given rule base.

Note that the location for inserting this rule in an existing rule base is subject to the rule processing logic contained in the existing rule base. Based on this logic, insert the rule after the rules that perform event processing, such as event de-duplication. But also remember to insert the rule before the rules that alter event contents or drop an event altogether.

This process might not be the optimal configuration for every implementation. For example, circumstances might exist that do not allow users to act on events at all, or allow them to act only on specific classes of events.

IMPORTANT: Manipulating and modifying an active T/EC rule base can have desired and undesired effects. The person making these changes should be a trained T/EC support person who understands the Operations Center software integration objectives for an implementation and the active rule base being modified.

10.5.3 Customizing T/EC for Integration with Operations Center software

This section provides instructions for preparing an environment to synchronize events between Operations Center software and T/EC consoles after installing the TecORB.

- "Creating the rb_Formula Rule Base in Tivoli" on page 269
- "Compiling the Rules" on page 269
- "Loading the Compiled Rules" on page 269
- "Restarting the T/EC Event Server" on page 270
- "Editing the Rule Base" on page 270
- "Modifying Classes" on page 271

NOTE: The name `rb_Formula` is an example name used in these sections to illustrate the procedures.

Creating the rb_Formula Rule Base in Tivoli

To create the rb_Formula Rule Base in Tivoli:

1. Verify with the Tivoli administrator that the environment has a current backup.
2. Right-click the *Event Server* icon, then click *Start-up*.

   If necessary, start the Event Server by right-clicking the *Event Server* icon, then clicking *Start-up*.
3. In the TME Desktop dialog box, right-click the Event Server for the T/EC slave, then click *Rules Bases* to open the Event Server Rule Bases dialog box.
4. Click the *Create* drop-down list, then click *Rule Base*.
5. Enter `rb_Formula` in the *Name* field.
6. Specify the correct host and path,

   or

   click *Directory* and select the path.
7. Click *Create & Close*.
8. To copy the contents of the latest running rule base, in the Event Server Rule Bases dialog box, right-click the latest running rule base icon, then click *Copy* to open the Copy Rule Base dialog box.
9. Select *rb_Formula* as the Destination rule base.
10. Select *Copy rules*, *Copy classes*, and *Overwrite files* in that destination.
11. Click *Copy & Close*.
12. Continue to "Compiling the Rules" on page 269 to compile and load the copied rules.

Compiling the Rules

To compile the rules in Tivoli:

1. In the Event Server Rule Bases dialog box, right-click the *rb_Formula* icon, then click *Compile* to open the Compile Rule Base dialog box.
2. Select *Trace Rules*.
3. Click *Compile* to compile the rules.
4. Ensure that no errors occur while the newly copied rules compile.

   If there are errors, notify the Tivoli administrator, as there might be a problem with the original rules.
5. Click *Close*.
6. Continue to "Loading the Compiled Rules" on page 269 to load the compiled rules.

Loading the Compiled Rules

To load the compiled rules in Tivoli:

1. In the Event Server Rule Bases dialog box, right click the rb_Formula icon, then click *Load*.
2. Click *Load & Close* to load the rules.

Restarting the T/EC Event Server

To restart the T/EC Event Server in Tivoli:

1. Shut down and restart the T/EC Event server to ensure that the rule base does not contain errors.
2. Click *Halt Server* to open the TME Desktop dialog box.
3. Right-click the EventServer icon, then click *Shut Down* to shut down the T/EC Event server.
4. Right-click again, then click *Start-up* to restart the T/EC Event server.

Editing the Rule Base

To edit the rule base in Tivoli:

1. Locate the `tec_forward.conf` file in the rule base directory.
2. Copy the `tec_forward.conf` file to a file named `formula.conf`.
3. Make a backup copy of the file.

   WARNING: Do not modify any existing filters.

4. Edit the ServerLocation, ServerPort, and TestMode properties in the `formula.conf` file as follows:

   ServerLocation: Operations Center server name or IP.

   ServerPort: Connection port for the server.

   TestMode: Indicates if this is a test.

   The following is an example of this file:

   ```
   #ServerLocation=NULL
   #TestMode=yes
   ServerLocation="Operations Center server name or IP"
   ServerPort=12345
   EventMaxSize=4096
   BufEvtPath=/etc/Tivoli/tec/Formula.cache
   #ConnectionMode=connection_oriented
   #Filter:Class=Logfile_Base
   ```

5. Compile and load the rules and restart the T/EC Event Server as described above.
6. Replace the following in the `/TEC_RULES/Formula_rules.rls` file:

Replace…	With…
`$which_postemsg`	The path to the location of postemsg.
`$new_rb_directory`	The directory path to the new rules base directory.

The following is an example of this file:

```
/*
The following rule processes TEC_Sync events from Operations Center
*/
rule:
tec_sync_from_master_tec:
(
description:'Processes TEC_Sync events from Operations Center server',
event: _event of_class 'TEC_Sync'
where [
originating_event_id: _orig_event_id,
new_status: _new_status
],
reception_action:
(
first_instance(
(
sprintf(_event_id, '%d%d%ld',[_event_handle,_server_handle,_date_reception]),
bo_set_slotval(_event,originating_event_id,_event_id),
re_send_event_conf('Formula',_event),
re_mark_as_modified(_event,_)
)
).
/*

 The following rule forwards updates to Operations Center
*/
change_rule:
update_status_on_master_tec:
(
description:'Forward alarm updates in TEC to Operations Center server',
event: _event of_class _class
where [
originating_event_id: _orig_event_id outside[0]
],
slot:status set_to _new_status outside[_status],
action:
(
/*EventID=(EventHandle)+(ServerHandle)+(DateReception)*/
exec_program(_event,'$which_postemsg',
'-f $new_rb_directory/TEC_RULES/Formula.conf -r HARMLESS
originating_event_id=%s new_status=%s TEC_Sync
TEC',[_orig_event_id,_new_status],'NO')
)
).
```

Modifying Classes

To modify the classes in Tivoli:

1. Add the synchronization classes to the rb_Formula rule base so that the superclass and class definitions in this and the following steps define the appropriate slot values.

   These slot values create the event uniqueness required to synchronize events between the two domains (Operations Center and T/EC).

2. Add the following entries to the top of the `/rb_Formula/TEC_CLASSES/ tec.baroc` file:

```
TEC_CLASS:
   TEC_Sync ISA EVENT
   DEFINES {
      new_status: STRING;
   };
   END
```

3 Modify the `root.baroc` file to add the string `originating_tec_hostname and originating_event_id` to the TEC_Class definition.

Add only the lines shown in bold; do not edit any other part of the file.

```
TEC_CLASS:
   EVENT
   DEFINES {
server_handle:  INTEGER,
  parse=no;
date_reception:  INT32,
  parse=no;
event_handle:  INTEGER,
  parse=no;
source:  STRING;
sub_source:  STRING;
origin:  STRING;
sub_origin:  STRING;
hostname:  STRING;
adapter_host:  STRING;
.
.
originating_tec_hostname: STRING, default = "$host";
       originating_event_id: STRING;
```

4 In the `root.baroc` file, replace `$host` with the local hostname of the T/EC server, then save the file.

This allows the TecORB to send TEC_Sync events when the event status changes.

The default `="$host"` statement must have the correct local hostname or IP address for the T/EC server. This is how the TecORB knows to which T/EC slave to send TEC_Sync events when event status changes occur in Operations Center software.

Compile and load the rules and restart the T/EC Event Server as described above.

10.5.4 Verifying that the Default Port is Available

The default port for TecORB is 1576. Prior to installing the TecORB, determine whether port 1576 is already in use on the T/EC server.

To verify port 1576 is available:

1 Enter one of the following commands at a command prompt:

Windows: `netstat -an | find "LISTEN" | find "1576"`

UNIX: `netstat -an | grep "LISTEN" | grep "1576"`

2 If above command results in a line ending with the word LISTEN, select a different port during installation.

See Installing TecORB on Windows.

10.5.5 Configuring TecORB

To configure TecORB:

1. If the TCP/IP port number 1576 is already in use on the T/EC server, select a different port.
2. To set the appropriate environment variables for the user who launches TecORB, perform one of the following steps:
   - For UNIX systems, run the /etc/Tivoli/setup_env.sh or /etc/Tivoli/setup_env.csh shell script.
   - For Windows systems, place wtdumper, wsetemsg, and postemsg in the user's path environment variables. These variables can be accessed through the system Control Panel.

10.5.6 Restricting Access by IP Address

It is possible to restrict access to some Operations Center components by IP address. By default, any host can connect to the TecORB. However, it is possible to specify the hosts that can connect to the ORB and thereby deny access to all other hosts.

To restrict access to TecORB:

1. Open the `tecorb.properties` file located in the TecORB configuration directory.
2. Locate the following line that normally starts the ORB:

   ```
   Command -Initial -Shared "directory_path_name -OAport 1572";
   ```

3. Add the following option to the code that initiates the ORB:

   ```
   -CORBA.Allow IP_Addresses
   ```

   where *IP_addresses* are those that are permitted to access the TecORB. Each IP address must be complete (no wildcards). Use commas to separate multiple host IP addresses.

   For example:

   ```
   Command -Initial -Shared "directory_path_name -OAport 1572 -CORBA.allow
   206.55.26.20,206.55.26.21,206.55.26.23
   ```

10.5.7 Starting and Stopping TecORB

- "Starting TecORB on UNIX" on page 274
- "Specifying the Startup Type for TecORB in Windows" on page 274
- "Manually Starting TecORB in Windows" on page 274
- "Manually Stopping TecORB" on page 274

Starting TecORB on UNIX

To start TecORB on Unix:

1. From the appropriate directory, at a command prompt, enter `tecorb`.

Specifying the Startup Type for TecORB in Windows

To specify the startup type on Windows:

1. From the desktop, click *Start* > *Settings* > *Control Panel* to open the Control Panel dialog box.
2. Double-click *Administrative Tools* to open the Administrative Tools dialog box.
3. Double-click *Services* to open the Services dialog box.
4. Right-click *NetIQ Operations Center*, click *TecORB*, then click *Properties* to open the Properties dialog box.
5. Click the *Startup Type* drop-down and select one of the following:

   Automatic: Starts TecORB automatically upon system startup.

   Manual: Starts TecORB manually.
6. Click *OK* to close the Properties dialog box.
7. Close the Services window.
8. Close the Control Panel window.

Manually Starting TecORB in Windows

To start TecORB in Windows:

1. Click *Start* > *Settings* > *Control Panel* to open the Control Panel dialog box.
2. Double-click *Administrative Tools* to open the Administrative Tools dialog box.
3. Double-click *Services* to open the Services dialog box.
4. Right-click *NetIQ Operations Center*, click *TecORB*, then click *Start*.

   The *Status* column changes to `Started`.

Manually Stopping TecORB

To stop TecORB:

1. Use the `mosstop` command.

10.6 NvORB for IBM Tivoli NetView

The NvORB wraps the IBM Tivoli NetView network management system with a CORBA object interface to enable Operations Center software to access many of NetView's core functions and data over a network. The Tivoli NetView adapter in Operations Center software connects to NvORB over the network (or locally, if NetView and Operations Center software are installed on the same

machine) to provide real-time access to managed objects on NetView from any Web browser through the Operations Center console. Use Operations Center software to automate and contain NetView elements in a centralized location.

For instructions on creating and configuring IBM Tivoli NetView adapters after the NvORB is installed, see Section 3.17, “IBM Tivoli NetView,” on page 114.

10.6.1 Preinstallation Notes

NvORB is installed as an extension to Tivoli NetView. When the first NetView user interface application starts on the NetView server, the NvORB process also starts and is ready for contact by the server. However, stopping the NetView user interface application also stops the NvORB process. This is a limitation of the NetView APIs. Operations Center software automatically reconnects when the NetView console is relaunched.

10.6.2 System Requirements

Table 10-11 outlines the system requirements for installing the NvORB:

Table 10-11 *System Requirements*

OS	Description
Windows	◆ Tivoli NetView 7.1.5 Server for Windows ◆ Windows 2000 ◆ Approximately 7 MB of available disk space on the drive where NetView is installed
AIX	◆ Tivoli NetView 7.15 Server for IBM AIX ◆ IBM AIX 5.2 or 5.3 ◆ Approximately 25 MB of available on the `/usr` file system
Solaris	◆ Tivoli NetView 7.15 Server for Solaris ◆ Solaris 2.9 ◆ Approximately 15 MB of available disk space on the `/usr/NV` file system

10.6.3 Verifying that the Default Port is Available

The default port for NvORB is 1572. Prior to installing NvORB, determine if port 1572 is already in use on the NetView server.

To configure ports for NvORB:

1. To verify port 1572 is available, enter one of the following at a command prompt:

   Windows: `netstat -an | find "LISTEN" | find "1572"`

   UNIX: `netstat -an | grep "LISTEN" | grep "1572"`

   If this command results in a line ending with `LISTENING`, then select a different port.

2. To select a port other than 1572, using any text editor, replace the port number in the `/usr/bin/registration/C/NvORB.arf` file.

   Use a port number above 1000 if running the NetView client as a user other than root.

10.6.4 Installing NvORB

The command to install NvORB depends on the operating system. In all cases, load the installation file from the `ORBS` directory on the Operations Center CD.

To install NvORB:

1. Verify NetView is not running.
2. Do one of the following:
   - On Windows NT, launch `\ORBS\NetView\NT\NetView-7.09.exe`
   - On AIX, issue the following command:

     `restore -qvxf /ORBS/NetView/AIX/nvorb_7.09.bff`
   - On Solaris, issue the following command:

     `pkgadd -d /ORBS/NetView/SolSparc/nvorb_7.09`

10.6.5 Configuring NvORB

Typically no configuration is required after installing NvORB. However, if the TCP/IP port number 1572 is already in use on the NetView server, then change a configuration file.

- "Setting the DISPLAY Environment Variable" on page 277
- "Restricting Access by IP Address" on page 277
- "Specifying a Map to Start NvORB" on page 277
- "Using NetView Process Flags" on page 278
- "Appending to the Log File When ORB Starts" on page 278

Setting the DISPLAY Environment Variable

On UNIX systems, set the DISPLAY environment variable to an available X server display.

To set the DISPLAY environment variable for bsh or ksh, use the following setting:

```
DISPLAY=<X server display>:0.0
export DISPLAY
```

To set the DISPLAY environment variable for csh, use the following setting:

```
setenv DISPLAY <X server display>:0.0
```

Restricting Access by IP Address

By default, any host can connect to the ORB. However, it is possible to specify the hosts that can connect to the ORB and thereby deny access to all other hosts.

To restrict access to NvORB:

1. Open the `/registration/C/NvORB.arf` file.
2. Locate the line that starts the NvORB, as shown:

   `Command -Initial -Shared "`*`directory_path_name`* `-OAport 1572";`

   where *directory_path_name* is the NvORB installation directory.
3. Append the following option to the code that initiates the ORB:

   `-CORBA.Allow` *`IP_addresses`*

   where *IP_addresses* are those that are permitted to access the NvORB.

   For example:

   ```
   Command -Initial -Shared "directory_path_name -OAport 1572 -CORBA.allow
   206.55.26.20,206.55.26.21,206.55.26.23
   ```

Specifying a Map to Start NvORB

When NetView starts, it loads a default map named `default`. The NvORB is set to run automatically when NetView starts, regardless of the map loaded at startup. However, it is possible to modify the registration file, `NvORB.arf`, to specify which map to load on startup.

IMPORTANT: If the map name specified in the `.arf` file and the map name given to NetView in the command line do not match, NvORB does not start.

To start NvORB only when it loads a specific map, add the following option to the `NvORB.arf` file:

`-map` *`map_name`*

where *map_name* is the name of the map to load at NetView startup.

You can abbreviate the `-map` command to `-m`. For example:

`Command -Initial -Shared` *`directory_path_name`* `NvORB -m Test -OAport 1572`

where *directory_path_name* is the NvORB installation directory.

Using NetView Process Flags

By default, NetView invokes NvORB at run time after selection of the appropriate menu item or executable symbol. Each selection loads and executes another instance of NvORB.

To change the behavior when NvORB starts:

1. Select and add one of three special flags to the command entry in the `NvORB.arf` file.
2. Enter the following at a command prompt in the `NvORB.arf` file:

   ```
   Command –Initial -Shared -Restart directory_path_name NvORB -m mapname -OAport
   1572
   ```

   where *directory_path_name* is the NvORB installation directory.

   The following explains the three switches:

   - **–Initial:** Instructs NetView to start NvORB when the NetView NNM starts.
   - **–Shared:** Instructs NetView to run only a single instance of the `NvORB` command at any time. In addition, this NvORB instance is shared, servicing multiple action requests. If another instance of NV starts, it attempts to start another instance of the ORB. This usually results in an ORB error because of a port conflict.
   - **–Restart:** Instructs NetView to restart NvORB automatically if the application stops while the NetView NNM is still running.

   For more information concerning NetView process flags and commands, see the documentation provided by NetView.

Appending to the Log File When ORB Starts

To append to the log file every time the ORB starts, add the command line argument `-LogAppend` to the `NvORB.arf` file. For example:

```
Command -Initial -Shared "/usr/OV/bin/NvORB -OAport 1572 -LogAppend"
```

10.6.6 Starting and Stopping NvORB

To start the NvORB:

1. Start the IBM/Tivoli TME 10 NetView Console by performing one of the following steps:
   - Enter `netview` at a command prompt.
   - Click the NetView Console icon in the `NetView` program folder.
2. Define an adapter in the Operations Center software to integrate with this NetView server.
3. Specify the NetView server's hostname and port number (1572 or the one chosen earlier).

To stop the NvORB, stop the NetView user interface, which also stops the NvORB.

A Adapter Property Reference

Configure adapter properties so that it can properly interface with its corresponding application or network management system, discovery tool, or trouble ticketing system. This section provides information regarding required adapter properties for each system:

A.1 Amazon Elastic Compute Cloud (Amazon EC2)

Table A-1 *Amazon EC2 Adapter Properties*

Property	Specify…
AWS Account Access Key ID	The active access key ID used to make secure requests to AWS.
AWS Account E-mail Address (informational)	The email address used to access the AWS account. This property is optional.
AWS Account ID	The AWS canonical user ID used exclusively for Amazon S3 resources such as buckets or files.
AWS Account Secret Access Key	The active secret access key used to make secure requests to AWS.
Alarm Columns	A comma-separated list that determines which alarm columns display and the order in which the alarm items display (source of alarm, alarm class, etc.) in the *Alarms* view. This property is not used by the Amazon Elastic Compute Cloud adapter as there are no alarms.
EC2 Poll Interval (minutes)	The interval, in minutes, that the adapter performs an automatic full refresh of the hierarchy. Defaults to 5.
Script.onError	A script that executes if the adapter fails for any reason. For example, the script can print the reason for the failure as `msg` using `log.info(msg)`.
Script.onInitialized	A script that executes upon adapter initialization. All of the `Script.*` properties are optional.
Script.onStarted	A script that executes when the adapter starts, either manually or automatically when the Operations Center server starts.
Script.onStopped	A script that executes after manually stopping the adapter.
Severity Mapping	Maps the Amazon EC2 severity codes to Operations Center severity codes. List the Amazon EC2 severity code first. The default is: `OK=OK:ALARM=CRITICAL:INSUFFICIENT_DATA=UNKNOWN`.
Show Community (Public) AMIs	Specify whether to show all community-shared Amazon Machine Images. If True, all community-shared AMIs are shown in the adapter. Defaults to False.

Property	Specify…
Show Community (Public) Snapshots	Indicates whether to show all community-shared snapshots. If True, all community-shared snapshots are shown in the adapter. Defaults to False

A.2 Blade Logic Operations Manager

Table A-2 *Blade Logic Operations Manager Adapter Properties*

Property	Specify…
Alarm Columns	A comma-separated list that determines which alarm columns display and the order in which the alarm items display (source of alarm, alarm class, etc.) in the *Alarms* view. This is user configurable.
Element Age Out Policy (minutes	The length of time, in minutes, to retain alarm elements. If no open alarms exist and the element's condition does not change in *n* minutes, and the element has no children, then the element disappears. If set to `0`, inactive elements timeout at 1 minute.
HierarchyFile	Set to a relative file name in the */OperationsCenter_install_path/*`database` directory. The file contains an XML description of hierarchy of elements that should be built below the element that represents the Blade Logic adapter. The default is `examples/BladeLogicOMHierarchy.xml`.
Script.onError	A script that executes if the adapter fails for any reason. For example, the script can print the reason for the failure as msg using log.info(msg).
Script.onInitialized	A script that executes upon adapter initialization. All of the `Script.*` properties are optional.
Script.onStarted	A script that executes when the adapter starts, either manually or automatically when the Operations Center server starts.
Script.onStopped	A script that executes after manually stopping the adapter.
Stylesheet File	The stylesheet file in the */OperationsCenter_install_path/*`database` directory. When this file is used, it is applied against the HierarchyFile as a style markup and produces the final output.
Use Alarm Times For Condition Changes	The date/time stamp to use for all alarm data stored by the Operations Center Data Warehouse. If `true`, the alarm's date/time stamp is used. If `false`, the date/time stamp of when the Operations Center server received the alarm is used. The default is `true`. Alarm history is stored based on the alarm time rather than alarm receipt time. Also, for SLA metric data based on alarm properties, the property values are recorded based on the alarm time instead of the alarm receipt time. Note that recording historical condition data for historical alarms is not supported.

A.3 BMC Remedy Action Request System (ARS)

Table A-3 *BMC Remedy ARS Adapter Properties*

Property	Specify…
Alarm Columns	A comma-separated list that determines which alarm columns display and the order in which the alarm items display (source of alarm, alarm class, etc.) in the *Alarms* view. A suggested list is: `assignedTo, description, status, lastModifiedBy, createDate, submitter` When adding a polled field to list, the column name must be prefixed with an underscore. For example, *_polledFieldName*.
Configuration File	The file containing BMC Remedy ARS configuration settings. The default is `RemedyConfiguration.xml`.
Hierarchy File	A relative file name in the */OperationsCenter_install_path/*`database` directory. The file contains an XML description of the hierarchy of elements built below the element that represents the adapter. The default is `examples/RemedyHierarchy.xml`.
Max Alarms Per Poll	The maximum number of alarms to retrieve per schema per poll period. The default is 500.
Max Alarms Per Schema	The maximum number of alarms that Operations Center actively maintains per Remedy Schema. Exceeding this number removes the oldest alarm. The default is 500.
Polling Interval	The number of seconds between queries for new tickets or re-queries for updating existing tickets. The default is 10.
Remedy Host Name	The hostname on which BMC Remedy ARS resides. This is a required property.
Remedy Password	The corresponding password for the provided user.
Remedy Server Port	The port number used by the Remedy ARS server. Configure this property only if Remedy Port-Mapper is not used and if the system uses version 7.*x* or later of the Remedy API JAR files.
Remedy User Name	The user name for the Remedy user account. This is a required property.
Script.onError	A script that executes if the adapter fails for any reason. For example, the script can print the reason for the failure as msg using log.info(msg).
Script.onInitialized	A script that executes when the adapter initializes.
Script.onStarted	A script that executes when the adapter starts, either manually or automatically when the Operations Center server starts.
Script.onStopped	A script that executes after manually stopping the adapter.
Show Query Info	If True, displays query statistics in the Operations Center log per schema, per poll period, after creating, updating, or deleting an alarm. The default is `false`.
Stylesheet File	The stylesheet file in the */OperationsCenter_install_path/*`database` directory that applies to the HierarchyFile as a style markup to produce the final output.

Property	Specify…
Use Alarm Times For Condition Changes	The date/time stamp to use for all alarm data stored by the Operations Center Data Warehouse. If `true`, the alarm's date/time stamp is used. If `false`, the date/time stamp of when the Operations Center server received the alarm is used. The default is `true`. Alarm history is stored based on the alarm time rather than alarm receipt time. Also, for SLA metric data based on alarm properties, the property values are recorded based on the alarm time instead of the alarm receipt time. Note that recording historical condition data for historical alarms is not supported.
Use String Pooling	If True, uses alarm and element property string pooling. The default is `true`.

A.4 BMC Software Event Manager

Table A-4 *BMC Software Event Manager Adapter Properties*

Property	Specify…
AlarmColumns	A comma-delimited list that determines which alarm columns display and the order in which the alarm items display (date/time, rule, etc.) in the *Alarms* view. The defaults are: `Status`, `Class`, `Description`.
ClosedAlarmsTimeout	The length of time, in seconds, to display an alarm after closing it in the Operations Center console. Enter -1 to display the alarm indefinitely. Enter 0 to remove it immediately. The default is `1800`.
ElementsTimeout	The length of time, in seconds, to retain alarm elements. If no open alarms exist and the element's condition does not change in n seconds, and the element has no children, then the element disappears. The element redisplays if another alarm is generated. The default is `300`. **ElementsTimeout < 0:** Never time out. **ElementsTimeout = 0:** Time out immediately. **ElementsTimeout > 0:** Time out after specified time expires.
HierarchyFile	A file in the */OperationsCenter_install_path/*`database` directory that contains an XML description of the element hierarchy to build below the adapter element. The default is `examples/MCHierarchy.xml`.
MaxAlarms	The maximum number of alarms that the adapter queries and retains. The default is 500.
Script.onError	A script that executes if the adapter fails for any reason. For example, the script can print the reason for the failure as msg using log.info(msg).
Script.onInitialized	A script that executes upon adapter initialization. All of the `Script.*` properties are optional.
Script.onStarted	A script that executes when the adapter starts.
Script.onStopped	A script that executes after manually stopping the adapter.

Property	Specify…
SeverityMapping	Maps the Event Manager severity codes to Operations Center severity codes. List the Event Manager severity code first. The default is: `FATAL=CRITICAL;CRITICAL=CRITICAL;WARNING=MAJOR;MINOR=MINOR; HARMLESS=INFORMATIONAL;UNKNOWN=UNKNOWN;INFO=INFORMATIONAL`
StylesheetFile	The stylesheet file in the */OperationsCenter_install_path/*`database` directory that applies to the HierarchyFile as a style markup to produce the final output.
UseAlarmTimesFor CondChanges	The date/time stamp to use for all alarm data stored by the Operations Center Data Warehouse. If `true`, the alarm's date/time stamp is used. If `false`, the date/time stamp of when the Operations Center server received the alarm is used. The default is `true`.

A.5 BMC Software PATROL

Table A-5 *BMC Software PATROL Adapter Properties*

Property	Specify…
Alarm Columns	A comma-separated list that determines which alarm columns display and the order in which the alarm items display (date/time, rule, etc.) in the *Alarms* view. The default list is: `ID,eid,status,class,host,appl,inst,description`
Hierarchy File	A file in the */OperationsCenter_install_path/*`database` directory that contains an XML description of the element hierarchy to build below the adapter element. The default is `examples/PatrolHierarchy.xml`.
KM Operations Permissions	ACL permissions to assign PATROL dynamic operations on elements. The default is `Define`.
Max Alarms Per Host	The maximum number of alarms that the adapter retains from each PATROL host. The default is 100. This total number of alarms is cumulative. For example, if the default is 200 and there are five hosts, then the adapter can accept 1000 total alarms. Alarms can also roll over. If there are 250 alarms from one host, the adapter can represent all of them, as long as the total does not exceed 1000.
Max Alarms Queried	The maximum number of events that each agent can mine and display in Operations Center. Assigning a low value on a very active set of agents can improve performance of initial discovery and does not impede the display of meaningful information. Alarms rapidly accumulate after connecting to the agent. The default is 50.
Max Hours Alarms Retained	The maximum number of hours to retain alarms. For example, enter 24 to retain only alarms received within the last 24 hours. Overrides the *Max Alarms Per Host* and *Max Alarms Queried* values.

Property	Specify…
Property Page Permissions	Assigns user access privileges for PATROL-specific property pages: Libraries, Communication Settings, Host, and Hosts (available at the host element level). The default privilege is `Define`, which allows users to edit and update all property pages. You must use one of the valid Operations Center access privileges: `View`, `Access`, `Manage`, or `Define`. For example, `Hosts=Define,Libraries=Define,CommunicationSettings=Define,Host=Manage` For details on these privileges, see the *Operations Center 5.5 Security Management Guide*.
Script.onError	A script that executes if the adapter fails for any reason. For example, the script can print the reason for the failure as msg using log.info(msg).
Script.onInitialized	A script that executes upon adapter initialization. All of the `Script.*` properties are optional.
Script.onStarted	A script that executes when the adapter starts.
Script.onStopped	A script that executes after manually stopping the adapter.
Show Timestamp in Agent System Output Window	If True, displays the time stamp of PATROL messages in the PATROL agent's System Output window which displays after clicking *Show Output Window* in the Operations Center console on a host machine. If False, it does not display the time stamp. The default is False. The time stamp values do not originate from PATROL; instead, Operations Center generates the time stamp when it receives a message. As a result, time stamps in Operations Center might not match exactly what displays in the PATROL console.
Stylesheet File	The stylesheet file in the */OperationsCenter_install_path/*`database` directory that applies to the HierarchyFile as a style markup to produce the final output.
Use Alarm Times For Condition Changes	The date/time stamp to use for all alarm data stored by the Operations Center Data Warehouse. If `true`, the alarm's date/time stamp is used. If `false`, the date/time stamp of when the Operations Center server received the alarm is used. The default is `true`. Alarm history is stored based on the alarm time rather than alarm receipt time. Also, for SLA metric data based on alarm properties, the property values are recorded based on the alarm time instead of the alarm receipt time. Note that recording historical condition data for historical alarms is not supported.

A.6 BMC Software PATROL Enterprise Manager

Table A-6 *BMC Software PATROL Enterprise Manager (PEM) Adapter Properties*

Property	Specify…
AccountMap	A file located in the */OperationsCenter_install_path/*`databases/examples` directory that constructs a map between Operations Center and Enterprise Manager users. The comments at the top of this file provide details on how to assemble the AccountMap file. The default is `CPAccountMap.properties`.

Property	Specify…
AckAffectsCondition	If True, an acknowledged alarm contributes to element condition and alarm counts. The default is `true`. For example, assume there is an element with one critical event `counts= 1-CRITICAL element=RED` Acknowledge the alarm and set AckAffectsCondition to `true`. This results in reducing the alarm count by one and changing the element condition to green: `counts= 0-CRITICAL element=GREEN` Acknowledge the alarm and set AckAffectsCondition to `false`. The result is the alarm still exists and the condition remains unchanged after acknowledging it: `counts= 1-CRITICAL element=RED`
AlarmAgeOutInSec	The length of time, in seconds, that an alarm should be retained. Alarms older than AlarmAgeOutInSec are aged out by a periodic worker thread that runs every minute. Upon adapter startup, all PEM events that are older than AlarmAgeOutInSec are ignored. All incoming events that are older than AlarmAgeOutInSec are also ignored. The default is 300.
AlarmAgeOutQueryExtDuration	Set to a value greater than zero to enable the *Query Alert History* right-click option on the PEM adapter element, which enables users to query historical alarms in the PEM persistent store. When selected, the *Query Alert History* option enables users to specify start and stop date/time boundaries for displaying historical alarms. Set this property to the length of time, in seconds, to retain the historical alarm window. Used in conjunction with the AlarmAgeOutInSec property, which must be set to a value greater than zero.
AlarmColumns	A comma-separated list that determines which alarm columns display and the order in which the alarm items display (date/time, rule, etc.) in the *Alarms* view. The default list is: `Tool ID, Pri, Ct, Desc, Cmnt, Ack, Acpt, Category, Operator, Description`
AlarmFilterOnStartup	Used to filter the alarms received from the PEM database. Contains the value used directly in the WHERE clause of two SQL statements that retrieve alarms from the PEM database. The query runs at adapter startup. The default is `state = 1`, which means that the default SQL statement executed against the database is `select * from alerts where state = 1` (where `1=CRITICAL`). The queries that run at adapter startup are: `select * from alerts where "+ AlarmFilterOnStartup +" order by timeReceived desc` `select * from alerts where "+ AlarmFilterOnStartup +" and alertId in ( " + reRetrieve + " )"` The property value must be a properly formatted WHERE clause of an SQL statement. For example, to filter by severity (in addition to state), set the property value to: `state=1` and `OSISeverity = X`. Restart the adapter after modifying this property. This property and AlarmFilterRuntime represent two different types of queries that can be executed from of the PEM adapter. Knowledge of the PEM database table structure is required. An improperly formatted WHERE clause of an SQL statement causes the adapter to not start.

Property	Specify…
AlarmFilterRuntime	Registers a filter/listener with PEM. The property value is used directly in WHERE clause of the query: create filter 0 from events, alerts where " + AlarmFilterRuntime. For more information about the Events and Alerts tables for PEM, see the *PEM Automation Guide*.
DBAccount	The user name for the CP Database. The default is `NetCmmnd`.
DBHost	The host location of the PATROL Enterprise Manager database. Since different parts of the PATROL Enterprise Manager system can reside on different servers, it is necessary to specify the database location.
DBPassword	The password used to access the PATROL Enterprise Manager database.
DBPort	The port number on which the database host listens. The default is 2043.
EHDHost	The host location of the PATROL Enterprise Manager Event Handler Daemon (EHD).
EHDPort	The port on which the PATROL Enterprise Manager Event Handler Daemon listens. The default is 3102.
EHDPullDuration	The length of time, in seconds, that the adapter directly queries the EHD. EHDPullOnTimeout must be set to True; otherwise, this property value is ignored. The default is 60 seconds.
EHDPullFrequency	Queries the EHD using the time interval specified as seconds. If EHDPullOnTimeout is set to True, the adapter enters a mode where it queries the EHD directly. It queries the EHD for the number of seconds specified in EHDPullDuration, at the time interval specified by EHDPullFrequency. The default is 10 seconds.
EHDPullOnTimeout	Set to True to have adapter query the EHD directly, using the duration and frequency specified by EHDPullDuration and EHDPullFrequency. Set to False (the default) to not use the direct query. If set to False, EHDPullDuration and EHDPullFrequency are ignored. Do not set to True unless experiencing problems where the EHD times out frequently.
EHDTimeout	The length of time to wait before Operations Center considers the EHD is down. This timeout elapses when there is no communication from the EHD. The default is 180 seconds.
ElementsTimeout	The length of time, in seconds, to age out elements. If no open alarms exist and the element's condition does not change in n seconds, and the element has no children, then the element disappears. The element redisplays if another alarm is generated. The default is 300 seconds. **ElementsTimeout < 0:** Never time out. **ElementsTimeout = 0:** Time out immediately. **ElementsTimeout > 0:** Time out after specified time expires.

Property	Specify…
FillnAlarmHistory	Set to True to include in the alarm history all events processed by PEM before the adapter connection is established. For example, if a PEM alert is created and closed while the adapter is unconnected to the PEM system, the alert is inserted into the alarm history when the adapter connects (provided the associated elements participated in the alarm profile). This guarantees that alerts and events processed by PEM while the adapter is not connected are delivered to Operations Center when the adapter reconnects. Every alert and event is received by the SMP server in the state and order that it was processed by PEM, albeit with a delay related to the time the connection was down. SMP receives the alert and then immediately receives the Close event. Set to False to include only the events that PEM processes while the adapter is connected.
HierarchyFile	A file in the */OperationsCenter_install_path/*`database` directory that contains an XML description of the element hierarchy to build below the adapter element. The default is `examples/CPHierarchy.xml`.
HostTokens	A list of PATROL Enterprise Manager token values that determines console connectivity. If Operations Center finds one of these tokens in an alarm, it uses the token to assign connectivity. The default is `null`.
MaxAlarms	The maximum number of alarms that the adapter queries and retains. The default is 0, which allows an unlimited number of alarms.
RelayServer	The name of the server on which the relay connection exists. For details on setting up a relay connection, see “Integration Using a Secure Relay Connection” on page 46.
RelayPort	The port number configured for use by the PEM adapter for relay communications.
RelaySecurity	The security level for the relay server: ssl or unsecured (meaning use cleartext, which is not case-sensitive).
Script.onError	A script that executes if the adapter fails for any reason. For example, the script can print the reason for the failure as msg using log.info(msg).
Script.onInitialized	A script that executes when the adapter is first initialized. All of the `Script.*` properties are optional.
Script.onStarted	A script that executes whenever the adapter is started.
Script.onStopped	A script that executes whenever the adapter is manually stopped.
ServicesFile	This property is relevant for administrators who want to have console capability. The default, `/etc/services`, is appropriate for UNIX systems. For Windows systems, copy the `/etc/services` file to a location on the Windows server, and change the value of this property to reference the location of that file.
StylesheetFile	The stylesheet file in the */OperationsCenter_install_path/*`database` directory that applies to the HierarchyFile as a style markup to produce the final output.
UseAlarmTimesFor CondChanges	The date/time stamp to use for all alarm data stored by the Operations Center Data Warehouse. If `true`, the alarm’s date/time stamp is used. If `false`, the date/time stamp of when the Operations Center server received the alarm is used. The default is `true`.

A.7 Castle Rock Computing SNMPc

Table A-7 *SNMPc Adapter Properties*

Property	Specify…
Alarm Columns	A comma-separated list that determines which alarm columns display and the order in which the alarm items display (date/time, rule, etc.) in the *Alarms* view. For SNMPc, the default is: `Current,CurrentPriority,SRCActual,SRCImplied,MessageText, VarValNth0,VarValNth1… VarValNth9`
Alarm Severity Mapping	Maps the SNMPc severity codes to Operations Center severity codes. List the SNMPc severity code first. The default is: `Critical=CRITICAL,Severe=CRITICAL,Major=MAJOR,Minor=MINOR,War ning=MINOR,Info=INFORMATIONAL,Normal=OK`
Alarms Discovery	Specify any combination of the following comma delimited values to filter and control discovered alarms: `MaxAlarms=[The max number of alarms to discover]` `BlockSize=[The number of alarms per discovery chunk]` `Severity=[SNMPc Severity String Value]` `BackTime=[The number of minutes to subtract from the current time to filter out old events]` MaxAlarms is a numeric value which represents the maximum number of alarms to discover. Specify any number less than 50000. If no value is specified, the overall Max Alarms property value is used. BlockSize is a numeric value that controls the number of alarms read per discovery block, The default is 1000, The maximum value is 5000. BackTime is a numeric value that represents the number of minutes to subtract from the current time. Events that are older than the calculated time are not included in the discovered set of alarms. The default is no back time limit. Severity is any valid SNMPc severity string value. Specify one or more severity settings. Only events with the specified severity are discovered, in the order listed. The valid severities are: CRITICAL, SEVERE, MAJOR, MINOR, WARNING, NORMAL, INFO. If no setting is specified, events of all severities are discovered. Use the other adapter properties to select a subset of alarms. For example, to discover no more than 5000 alarms, set the maximum BlockSize to 100, To exclude events older than 24 hours and to discover CRITICAL, SEVERE, MAJOR, MINOR, WARNING, and INFO alarms, use the following property settings: `MaxAlarms=5000,BlockSize=100,BackTime=1440,Severity=CRITICAL, Severity=SEVERE,Severity=MAJOR,Severity=MINOR,Severity=WARNIN G, Severity=INFO`
Discovery Timeout Interval	By default, adapter discovery times out after 30 seconds.Set to a number greater than 30 to increase the timeout interval in seconds. If this property is left empty or set to a number less than 30, the default is used.
Hierarchy File	A file in the */OperationsCenter_install_path/*`database` directory that contains an XML description of the element hierarchy to build below the adapter element. The default is `examples/DefaultSnmpcHierarchy.xml`.

Property	Specify…
Max Alarms	The maximum number of alarms that the adapter queries and retains. The default is 500. Set to 0 to allow an unlimited number of alarms.
SNMPc Login Password	The password for the user name supplied in SNMPc Login User ID.
SNMPc Login User ID	The user ID used to log in to the SNMPc server.
SNMPc Server IP Address	The IP address of the server where SNMPc is installed.
Script.onError	A script that executes if the adapter fails for any reason. For example, the script can print the reason for the failure as `msg` using `log.info(msg)`.
Script.onInitialized	A script that executes when the adapter initializes. All of the `Script.*` properties are optional.
Script.onStarted	A script that executes when the adapter is starts.
Script.onStopped	A script that executes after manually stopping the adapter.
StylesheetFile	The stylesheet file in the */OperationsCenter_install_path/*`database` directory that applies to the HierarchyFile as a style markup and produces the final output.
Use Alarm Times For Condition Changes	The date/time stamp to use for all alarm data stored by the Operations Center Data Warehouse. If `true`, the alarm's date/time stamp is used. If `false`, the date/time stamp of when the Operations Center server received the alarm is used. The default is `true`. Alarm history is stored based on the alarm time rather than alarm receipt time. Also, for SLA metric data based on alarm properties, the property values are recorded based on the alarm time instead of the alarm receipt time. Note that recording historical condition data for historical alarms is not supported.

A.8 Cisco Info Center

Table A-8 *Cisco Info Center Adapter Properties*

Property	Specify…
AccountMap	The file used to construct a map between Operations Center and CIC users. Comments at the top provide details on how to assemble the AccountMap file. The default is `examples/CICAccountMap.properties`.

Property	Specify…
AckAffectsCondition	If True, an acknowledged alarm contributes to element condition and alarm counts. The default is `true`. For example, assume there is an element with one critical event `counts= 1-CRITICAL element=RED` Acknowledge the alarm and set AckAffectsCondition to `true`. This results in reducing the alarm count by one and changing the element condition to green: `counts= 0-CRITICAL element=GREEN` Acknowledge the alarm and set AckAffectsCondition to `false`. The result is the alarm still exists and the condition remains unchanged after acknowledging it: `counts= 1-CRITICAL element=RED`
AlarmColumnDefinition	The file used to define alarm columns, The default is `examples/NetcoolAlarmColumns.properties`.
AlarmColumns	A comma-separated list that determines which alarm columns display and the order in which the alarm items display (date/time, rule, etc.) in the *Alarms* view. For CIC, the default is: `Node, Count=Tally, Acknowledged, Description=Summary`
AlarmSeverityMapping	Maps the Cisco Info Center severity codes to Operations Center severity codes. List the Cisco Info Center severity code first. The default is: `CLEAR=OK,INDETERMINATE=INFORMATIONAL,WARNING=MINOR, MINOR=MINOR,MAJOR=MAJOR,CRITICAL=CRITICAL`
CheckConnection	If True, polls on an active basis to check for connection to CIC server. The default is True.
ElementsTimeout	The length of time, in seconds, to age out elements. If no open alarms exist and the element's condition does not change in n seconds, and the element has no children, then the element disappears. The element redisplays if another alarm is generated. The default is 300 seconds. **ElementsTimeout < 0:** Never time out. **ElementsTimeout = 0:** Time out immediately. **ElementsTimeout > 0:** Time out after specified time expires.
HierarchyFile	A file in the */OperationsCenter_install_path/*`database` directory that contains an XML description of the element hierarchy to build below the adapter element. The default is `examples/CICHierarchy.xml`.
HostTokens	A list of CIC token values that determine console connectivity. If Operations Center finds one of these tokens in an alarm, it uses the token to assign connectivity. The default is `Node`.
IDUCTimer	The time interval used to check for updates for a specific Cisco Info Center integration. Overrides the Netcool global IDUC timer setting. The default is `60`. Valid values are greater than 5 seconds.
MaxAlarms	The maximum number of alarms that the adapter queries and retains. The default is 0, which allows an unlimited number of alarms.
ObjectServerAccount	The user name for connecting to the object server. The default is `root`.

Property	Specify…
ObjectServerHost	The host location of the object server.
ObjectServerName	The name of the object server. The default is `INFOSERVER`.
ObjectServerPassword	The password for the user name supplied as the ObjectServerAccount.
ObjectServerPort	The port on which the object server host listens. The default is 4100.
ObjectServerVersion	The version of Object Server software that is installed.
OperationsMenu	The CIC object server menu table used to populate the right-click operations for CIC alarms. The default is to leave blank, which then uses the default AlertsMenu CIC menu. Specify a different menu table name and the adapter attempts to use that menu table. If it fails, no menu operations display and a warning is logged in the `formula.trc` file.
RelaySecurity	The security level for the relay server: SSL or unsecured (meaning use cleartext, which is not case-sensitive). The default is `SSL`.
RelayServer	The name of the server on which the relay connection exists. For details on setting up a relay connection, see “Integration Using a Secure Relay Connection” on page 107.
RelayServerPort	The port number configured for use by the Cisco Info Center adapter for relay communications.
Script.onConnected	A script that executes when the adapter successfully connects to the object server. The default is: `@adapters/CIC/setServerSkewTime.fs` This script calculates the time differential (skew) between the Operations Center and object servers. This skew then affects calculations of the dates or time stamps for user presentation and information logging/stamping. If the Operations Center server runs on the same machine as the CIC server, then there is no skew to calculate. In this case, the time skew script is unnecessary. If a site synchronizes the time for its servers using the standard UNIX timeserver service (usually port 37), then the difference in time between the Operations Center server and the machine running the CIC server might be negligible. In this case, ignore this parameter. The supplied script calculates the difference in time between the Operations Center server and the target CIC server by running the REXEC protocol to invoke the `date` command on the UNIX host. If the CIC server runs on a different operating system, or if a site has restricted the access to the REXEC protocol, use a different mechanism to calculate the difference in time between the machine running Operations Center and the CIC server. If using a different mechanism, study the supplied script as a model for making adjustments to the Operations Center CIC adapter. The script invokes the adapter method `setServerTimeSkew( )` using the number of milliseconds (seconds multiplied by 1000) of difference between the Operations Center server and CIC server.
Script.onDisconnected	A script that executes when the adapter disconnects from the object server.
Script.onError	A script that executes if the adapter fails for any reason. For example, the script can print the reason for the failure as msg using log.info(msg).

Property	Specify…
Script.onInitialized	A script that executes when the adapter initializes. All of the `Script.*` properties are optional.
Script.onStarted	A script that executes when the adapter is starts.
Script.onStopped	A script that executes after manually stopping the adapter.
StylesheetFile	The stylesheet file in the */OperationsCenter_install_path/*`database` directory that applies to the HierarchyFile as a style markup and produces the final output.
UseAlarmTimesFor CondChanges	The date/time stamp to use for all alarm data stored by the Operations Center Data Warehouse. If `true`, the alarm's date/time stamp is used. If `false`, the date/time stamp of when the Operations Center server received the alarm is used. The default is `true`.

A.9 CiscoWorks2000 DFM

Table A-9 *CiscoWorks2000 DFM Adapter Properties*

Property	Specify…
Alarm Columns	A comma-separated list that determines which alarm columns display and the order in which the adapter-specific alarm items display (source of alarm, alarm class, etc.) in the *Alarms* view. A suggested list is: `Create Time,Class Name,Instance Name,Event Name,Event Type,Message Type,Message String,Domain Manager,Parameter,Count`
Broker	The remote broker instance that allows communication with the CiscoWorks Domain Manager. Specify *`hostname`*`:`*`port`*.
Broker Password	The password corresponding to the Broker User.
Broker User	The user name that CiscoWorks uses to communicate with DFM. This is not the same as the CiscoWorks GUI login user. This is the user name prompted for during the DFM server install. This is a required property.
Class List	The initial list of classes for discovery. The default list is: `Host,Hub,MSFC,Probe,Router,RSFC,RSM,Switch,Undiscovered,VLAN`
Connection Attempts Before Failover	The maximum number of times a connection attempt is made before it fails over to the secondary CiscoWorks server. The default is 3.
DM Password	The password to log in to the Domain Manager.
DM User	The user name to log in to the Domain Manager.
Depth of Initial Exploration	The number of element levels to explore during initial discovery. Defaults to 8.
Display Name	Specify the display name for the adapter.
Domain List	A comma delimited list of domains to access.
Domain Manager	The name of the Domain Manager. Only one Domain Manager is supported per adapter instance. The default is `dfm`.

Property	Specify…
Explore Relations	A comma delimited list of all relations to return. Defaults to `ConsistsOf, ConnectedVia`.
Filter Alarms by Class List	If True, displays only those notifications (and their root cause alarms/notifications) that are members of the classes specified in the Class List property. Defaults to `False`.
Hide Relationships	If True, relationship elements such as consists of and member of do not display in the Operations Center console. Defaults to False.
Notification List Profile	Specify the notifications to include; the default is `ALL_NOTIFICATIONS`.
Notification Mapping	Changes the default mapping of DFM notification types to Operations Center alarm severities. Uses the defaults if this property is left blank. Valid Event types include: MR_AGGREGATION, MR_CASUALITY, MR_EVENT, MR_IMPORTED_EVENT, MR_PROBLEM, MR_PROPAGATED_AGGREGATION, MR_SAMETYPE, MR_SYMPTOM. Valid Operations Center severities are: CRITICAL, INFO, MAJOR, MINOR, OK, IGNORE. For example, to set symptoms to INFORMATIONAL severity and problems to MAJOR severity, enter: `MR_EVENT=info;MR_PROBLEM=major`
Script.onError	A script that executes if the adapter fails for any reason. For example, the script can print the reason for the failure as msg using log.info(msg).
Script.onInitialized	A script that executes when the adapter initializes. All of the `Script.*` properties are optional.
Script.onStarted	A script that executes when the adapter starts, either manually or automatically when the Operations Center server starts.
Script.onStopped	A script that executes after manually stopping the adapter.
Secondary Broker	Part of a secondary CiscoWorks server definition, for use in a failover scenario. See the definition for Broker above.
Secondary Broker Password	Part of a secondary CiscoWorks server definition, for use in a failover scenario. See the definition for Broker Password above.
Secondary Broker User	Part of a secondary CiscoWorks server definition, for use in a failover scenario. See the definition for Broker User above.
Secondary DM Password	Part of a secondary CiscoWorks server definition, for use in a failover scenario. See the definition for Broker Password above.
Secondary DM User	Part of a secondary CiscoWorks server definition, for use in a failover scenario. See the definition for DM User above.
Use Event Instance Name	Determines the CiscoWorks event field used to roll up incoming CiscoWorks alarms, populate the alarm key, and increment the alarm's `Count` field. Set to `false` to use the value of the CiscoWorks *ElementName* property. Set to `true` to use the value of CiscoWorks *Name* property. Defaults to `false`.

A.10 Computer Associates Spectrum

Table A-10 *Computer Associates Spectrum Adapter Properties*

Property	Specify…
Alarm Columns	A comma-separated list that determines which alarm columns display and the order in which the alarm items display (date/time, rule, etc.) in the *Alarms* view. The default list is: `modelName,Ack=Acknowledged,description` Ask the Spectrum Administrator to determine which the columns are available. Specify each column name using the correct case.
CORBA Callback Port	Specifies the port used by the SPECTRUM OSAgent to perform callback communications to the adapter. Specify a value between 1 and 65535. The default is 1581.
Call Performance Logging	If True, this debug option causes the Spectrum Integration to note the timing and context of each call it makes to the SpectroServer. This information is valuable when attempting to optimize performance on larger Spectrum installations. The default is False.
Connection Retry Delay	**Applicable for Spectrum version 9.0 adapters only.** The minimum delay (in seconds) between “keep alive” connection attempts to the Spectrum server. Default is `30`.
Device Extra Relations	A comma-separated list of the hex numbers of additional relationships to display for device-class objects. The Spectrum Integration defines a set of relationships that are commonly used for devices. The Spectrum administrator can define additional relationships. Also, unusual relationships might be in use in an environment. If children exist in a relationship listed here, the model contains the generated children.
Exclude Model Class	Specify the numeric value of one or more Model Classes to filter out alarms or elements. Separate values with a comma. For example, to exclude any Spectrum elements or alarms with a Model Class of *Port*, type `15`.
Follow Remote Links	If True, the Spectrum Integration attempts to follow remote model links. These links lead from one SpectroServer to another, in the distributed SpectroServer configuration. Additional required steps for configuring SpectroServer security to fully implement this feature are described in “Understanding Spectrum Adapter Features” on page 66. The default is `true`.
Model Domain ID to IP Address Map	**Applicable for Spectrum version 8.*x* adapters only.** The IP address of a specific model domain that is communicated to the Spectrum Integration. This directly specifies a model domain IP address if it cannot reach a particular model domain through the common lookup mechanism. This comma-separated list uses the following format, where *0xnnnnnnnn* is the model domain identifier and *y.y.y.y* is the IP address of the hosting SpectroServer: *`0xnnnnnnnn=y.y.y.y`*

Property	Specify…
Model Name to Properties Files Map	**Applicable for Spectrum version 8.*x* adapters only.** The model names and their associated property files to load. By default, Operations Center loads the `CsStandard.30` file, which displays in the main element property pages. The format is: *`model_name=one_or_more_filenames_separated_by_commas`* Model names are separated by semicolons: *`model_name=file1, file2, fileN;model_name=file1`* The default is: `Gen_IF_Port=CsMonMdlIn.30,CsMonMdlOut.30` This creates two additional element properties pages entries for elements of model type: `Gen_IF_Port : CsMonMdlIn.30 and CsMonModlOut.30` To determine available files, search the directory of model names in *`Spectrum_home`*`/SG-Support/CsGib`, and add the file names and the model types to the adapter property. It is necessary to look through the models of interest and identify the data to display.
Name Service Port	The port number on which the Spectrum CORBA Name Service is configured to listen. The default is 14006.
Non-Device Extra Relations	A comma-separated list of the hex numbers of additional relationships to display for non-device-class objects. The Spectrum Integration defines a set of relationships that are commonly used for non-devices. The Spectrum administrator can define additional relationships. Also, unusual relationships might be in use in an environment. If children exist in a relationship listed here, the model contains the generated children.
ORB Init Parameters	Use this property only under the direction of Support (http://www.netiq.com/support/).
ORB Timeout	The length of time, in seconds, to wait for a SpectroServer response to calls. Use this property only under the direction of Support (http://www.netiq.com/support/). The default is 60.
OSAgent	The IP address or domain name of a machine that is running an OSAgent to use if the integration cannot locate and default to an initial OSAgent. In a conventional installation, this is the machine on which the SpectroServer runs.
Polling	Polls the remote agents to check for connection. The default is `false`. Set this property to `true` when `Show Global Collections` is enabled.
Script.onError	A script that executes if the adapter fails for any reason. For example, the script can print the reason for the failure as msg using log.info(msg).
Script.onInitialized	A script that executes upon adapter initialization. All of the `Script.*` properties are optional.
Script.onStarted	A script that executes when the adapter starts, either manually or automatically when the Operations Center server starts.
Script.onStopped	A script that executes after manually stopping the adapter.

Property	Specify…
Show Global Collections	**Applicable for Spectrum version 9.0 adapters only.** If true, the Spectrum adapter includes global collections in the *Elements* hierarchy tree. The default is `false`. The `Polling` property must be set to `true` when this feature is enabled.
SpectroServer	The CORBA name of the SpectroServer, which is almost always the same as the short name of the system on which the SpectroServer is running. For example, if the SpectroServer runs on qa4sun0.mosol.com, the SpectroServer name is qa4sun0. Verify the CORBA name by running the `osfind` command in the SpectroServer installation.
Spectrum Cause File Directory	**Applicable for Spectrum version 9.0 adapters only.** The Spectrum Cause Files map the Spectrum cause id (for alarms) to the desciption of the alarms cause. Specify the Operations Center directory where you have saved copies of these files from your Spectrum installation at *`SpectrumInstall`*`/SG-Support/CsPCause` at integration setup.
Spectrum Installation Directory	**Applicable for Spectrum version 8.*x* adapters only.** The Spectrum installation directory. Enables the integration to gather additional information about properties of the Spectrum Models. If the Operations Center server is not on the same system as the SpectroServer, consider copying the files to the Operations Center server. The SpectroServer does not need to run; however, the files must be available. Another option is to set up a remote file link. The default is `/opt/spectrum`.
Spectrum OneClick Topology Config Directory	**Applicable for Spectrum version 9.0 adapters only.** Location of the XML files needed to retrieve the properties to be consistent with the Spectrum OneClick client. Defaults to `/SPECTRUM/tomcat/webapps/spectrum/WEB-INF/topo/config`. Replace the default the Operations Center directory where you have saved copies of these files from your Spectrum installation at *`SpectrumInstall`*`/tomcat/webapps/spectrum/WEB-INF/topo/config` at integration setup.
Spectrum User	The user names added to the Spectrum `.hostrc` files on the SpectroServers to be connected. The SpectroServer does not permit Operations Center to connect to it unless the Spectrum user name matches an entry in the .hostrc file.
Universe	**Applicable for Spectrum version 8.*x* adapters only.** Do not modify this internal setting.
Use Default Algorithm for Global Collections Status	**Applicable for Spectrum version 9.0 adapters only.** If `true`, status information of children is propagated to the Global Collections parent folders. If `false`, no status is propagated and parent folders remain in the `unknown` state.The default is `false`.

A.11 Computer Associates (CA) Unicenter

Table A-11 *Computer Associates (CA) Unicenter Adapter Properties*

Property	Specify…
AlarmColumns	A comma-separated list that determines which alarm columns display and the order in which the alarm items display in the *Alarms* view. Add the Node keyword to the beginning of the column, as in the following example: `Node,Ack,Ann,Number,Console,Description` The possible alarm columns are: *Console, Attrib, Type, Ack, Ann, TimeGen, MsgNum, Exit, Node, User, Station, Number, DuplicateIDs, Description, Device, Workload, PInfo, UserData, Category, Source, Tag*. This property might seem redundant when considering alarms generated from objects below the server level in Unicenter. But in some cases, it is the only way to identify the hostname of the source of an alarm directly from the *Alarms* view.
AutoShowDSM	If True, automatic discovery of DSM level elements occurs when the adapter starts. If False, it is necessary to manually discover DSM level elements by using the *Details* > *Show Operation* option. The default is False.
Command.Ping	The command used for the `Ping` function. The syntax varies among operating systems. The default is `ping -t`.
Command.TraceRoute	The command used for the `TraceRoute` function. The syntax varies among operating systems. The default is `tracert`.
EventManagerHosts	The hostnames of the machines on which Unicenter is installed. The default is `localhost`. Set this property to one or more comma-separated hostnames that instruct the adapter where to look for Event Manager (EM) consoles. The following values are relative to the location of the ORB itself, not the Operations Center server: ◆ **localhost:** A special hostname used if the UniORB runs on the same server as a Unicenter server. This causes the ORB to access the Unicenter log files using the local path name `x:/NSM/LOGS` instead of a UNC name. ◆ **fully.qualified.hostname:** Use this value if the UniORB needs to map a UNC name to access Unicenter log files, located on `\\hostname\UNISHARE$`. If the Unicenter Enterprise Managers reside on different machines than the Unicenter World-View and the UniORB, set the EventManagerHosts adapter property to the machines on which the Unicenter Enterprise Manager is installed. If this property is blank, the adapter searches for EM consoles on all of the hosts that Unicenter objects reference in their DSM_Address property.
MaxAlarmAgeHours	The number of hours that Operations Center performs initial alarm retrieval. Set to a larger number, such as 72. This provides three days worth of event messages. If the message count climbs too high during retrieval, lower the number. Coordinate with the MaxAlarms property. The default is 12.
MaxAlarms	The maximum number of alarms that the adapter queries and retains. The default is 5000. Larger customers might want to increase this value to 20,000–50,000 to obtain messages that cover the previous three days or more. Coordinate with the MaxAlarmAgeHours property.

Property	Specify…
MineTNGIcons	The default is False, meaning no icons are mined. If True, icons are mined from Unicenter and copied into the */OperationsCenter_install_path*/html/images/large and */OperationsCenter_install_path*/html/images/small directories. All adapters use the icons in the .../html/images directory.
OperationTimeout	The amount of time, in milliseconds, that an operation initiated from Operations Center waits for a reply. The default is 60000.
ReadOnlyConnection	Enables operations for users who have a read-only connection. If True, disables all operations that affect the adapter state. If False, no restrictions apply. The default is False.
RepositoryName	The name of the database repository used by Unicenter. This is a case-sensitive value. Set it to the exact value on the menu that Unicenter displays when opening applications that access the repository. By default, when SQL Server is the repository for Unicenter, the RepositoryName is the uppercase name of the server on which the repository is installed. Set in conjunction with RepositoryPassword.
RepositoryPassword	The password for the repository named in RepositoryName. This is a case-sensitive value.
RepositoryUser	The user name used to access the Unicenter repository. The default is sa.
Script.onError	A script that executes if the adapter fails for any reason. For example, the script can print the reason for the failure as msg using log.info(msg).
Script.onInitialized	A script that executes when the adapter initializes. All of the Script.* properties are optional.
Script.onStarted	A script that executes when the adapter starts, either manually or automatically when the Operations Center server starts.
Script.onStopped	A script that executes after manually stopping the adapter.
ServerHost	The fully qualified TCP/IP hostname of the server on which UniORB is installed.
ServerPort	The TCP/IP port number on which UniORB runs, as specified in the Control Panel applet. The default is 1580.
Polling	If true, polls to retrieve events in batches. If false, the adapter listens for events. The default is false.

A.12 EMC SMARTS

Table A-12 *EMC SMARTS Adapter Properties*

Property	Specify…
Alarm Columns	A comma delimited list that determines which alarm columns display and the order in which the adapter-specific alarm items display (source of alarm, alarm class, etc.) in the *Alarms* view. A suggested list is: `Create Time,Class Name,Instance Name,Event Name,Event Type,Message Type, Message String,Domain Manager,Parameter,Count.`
Broker	The remote broker instance that enables communication with the SMARTS Domain Manager. Specify the hostname and port. The default is localhost:426.
Broker Password	The password to log in to the remote broker instance. The default is `admin`.
Broker User	The user name to log in to the remote broker instance. The default is `admin`.
Class List	A list of classes for discovery in forming the root of the SMARTS hierarchy. Set to InChargeDomain to view all classes in all domains. To discover specific classes for all domains, specify `InChargeDomain=Class1,Class2`. Available classes include `Host,Switch,Router,MSFC`. Using regular expressions, apply additional filtering to class and domain. For example, `InChargeDomain=Switch=NYC.*,MSFC`.
Connection Attempts Before Failover	The maximum number of times a connection attempt is made before it fails over to the secondary SMARTS server.
DM Password	The password to log in to the SMARTS Domain Manager. The default is `admin`.
DM User	The user name to log in to the SMARTS Domain Manager. The default is `admin`.
Depth of Initial Exploration	The number of element levels to explore during initial discovery. Defaults is 8.
Display Name	EMC SMARTS property used to mine element names. Defaults is `DisplayName`.
Domain List	A comma delimited list of domains to access that are under control of the domain manager specified in the Domain Manager property. If a domain isn't subscribed to by the specified Domain Manager, Operations Center won't be able to access it.
Domain Manager	The name of the SMARTS Domain Manager. Only one Domain Manager is supported per adapter instance. The default is `DFM`.
Explore Relations	A comma delimited list of all relations to return. Defaults to `ConsistsOf, ConnectedVia`.
Filter Alarms by Class List	If True, displays only those notifications (and their root cause alarms/notifications) that are members of the classes specified in the Class List property. Defaults to `False`.
Hide Relationships	If `True`, relationship elements such as consists of and member of do not display in the Operations Center console. Defaults to `false`.
ICS Domains	A comma delimited list of any domains in your EMC SMARTS environment that can send out syslog and SNMP trap events. This is used to identify and organize these type of events in the Elements hierarchy.

Property	Specify…
Notification List Profile	The notification list to subscribe to for changes and removal of notifications. The default is `ALL_NOTIFICATIONS`.
Notification Mapping	A customized list of InCharge notification types mapping to Operations Center alarm severities. Uses defaults if property is blank. Specify each mapping as `NotificationType=AlarmSeverity`. For example, `MR_EVENT=info;MR_PROBLEM=major`. Event notification types include `MR_AGGREGATION, MR_CASUALITY`, `MR_EVENT`, `MR_IMPORTED_EVENT`, `MR_PROBLEM`, `MR_PROPAGATED_AGGREGATION`, `MR_SAMETYPE`, and `MR_SYMPTOM`. Valid Operations Center severities are: `CRITICAL`, `INFO`, `MAJOR`, `MINOR`, `OK`, and `IGNORE`.
Script.onError	A script that executes if the adapter fails for any reason. For example, the script can print the reason for the failure as msg using log.info(msg).
Script.onInitialized	A script that executes when the adapter initializes.
Script.onStarted	A script that executes when the adapter starts, either manually or automatically when the Operations Center server starts.
Script.onStopped	A script that executes after manually stopping the adapter.
Secondary Broker	Part of a secondary SMARTS server definition, for use in a failover scenario. See the definition for Broker above.
Secondary Broker Password	Part of a secondary SMARTS server definition, for use in a failover scenario. See the definition for Broker Password above.
Secondary Broker User	Part of a secondary SMARTS server definition, for use in a failover scenario. See the definition for Broker User above.
Secondary DM Password	Part of a secondary SMARTS server definition, for use in a failover scenario. See the definition for DM Password above.
Secondary DM User	Part of a secondary SMARTS server definition, for use in a failover scenario. See the definition for DM User above.
Secondary Domain Manager	Part of a secondary SMARTS server definition, for use in a failover scenario. See the definition for Domain Manager above.
Use Event Instance Name	Determines the SMARTS event field used to roll up incoming SMARTS alarms, populate the alarm key, and increment the alarm's `Count` field. Set to `false` to use the value of the SMARTS *ElementName* property. Set to `true` to use the value of SMARTS *Name* property. Defaults to `false`.

A.13 HP OpenView Network Node Manager

Table A-13 *HP OpenView Network Node Manager Adapter Properties*

Property	Specify…
Adapter Instance	A pre-3.5 OpenView adapter instance to ensure backward compatibility. Determine the adapter instance from the adapter DName. For example, in `openview:4=MyOV/root=Elements`, the adapter instance is 4.

Property	Specify…
Adjust Historical Varbinds	If True, the first two varbinds are removed from SMNPv2 protocol events at historical event load time (when the NNM adapter first starts). If False, no varbinds are removed from SNMPv2 protocol events. Regardless of the property setting, after the initial historical event load, no varbinds are removed from SNMPv1 protocol events nor are they removed from any real-time NNM events. The default is False.
Alarm Columns	A comma-separated list that determines which alarm columns display and the order in which the alarm items display (date/time, rule, etc.) in the *Alarms* view. The default list is: `Ack,EventTime,EventSource,EventName,Message` Aliases for Alarm Columns: Assign new names to alarm columns using the format: *`display_name=current_name`*. For example, `Note=Message` displays messages in a column named *Note*.
Application Name	The default is `Formula`.
Auto Discovery	If True, discovers all Symbols at adapter startup. If False, discovers at adapter startup only the Symbols in the levels specified in the Discovery Depth property. The default is False.
Condition Mapping	Maps one or more OpenView conditions to Operations Center severity codes. The default is to leave it blank, which then uses the default mappings. To define a particular code mapping, use the following format, separating each mapping with a semicolon: `Openview=Formula;Openview=Formula;` For example, to change the condition mapping of Marginal nodes within OpenView to INFO, and to change user1 to CRITICAL, use the following ConditionMapping value: `Marginal=Info;user1=Critical;` Table A-14 shows the default mappings between OpenView and Operations Center, and Table A-15 lists possible Operations Center code mapping values. Value comparisons are case sensitive, so Marginal is different from marginal.
Connection Verbose Logging	If True, logs reconnect attempts to OVW.
Default Event Fields for Event Normalization	The default event fields used when the Event Normalization setting Affected Element is derived from equals Object Attribute. The default is `{EventSource}/{EventID}`.
Default Object Attribute for Event Normalization	The default object attribute used when the Event Normalization setting Affected Element is derived from equals Object Attribute. The default is `snosIdentifyingInfo`.
Discovery Depth	If the Auto Discovery property is False, Symbols in the levels specified in this property are discovered at adapter startup. If the Auto Discovery property is True, all Symbols are discovered at adapter startup. The minimum value is 1.
Hierarchy File	A file in the */`OperationsCenter_install_path`/database* directory that contains an XML description of the element hierarchy to build below the adapter element. The default is `examples/DefaultNNMHierarchy.xml`.
Map Name	A map name that restricts adapter access to only those OVW sessions that have the map open. If no map is specified, the adapter can connect to any OVW session. The default is `default`.

Property	Specify…
Max Alarms	The maximum number of alarms that the adapter queries and retains. Enter 0 to allow an unlimited number of alarms. The default is 500.
Notify Operation Completed	If True, notifies the user when the Clear All Alarms or Reload All Alarms operations finish. The default is True.
ORB Connection Check Delay	Check s the connection to the OvORB at the specified time interval (in seconds). If the connection is down, all alarms are removed from the adapter. Repopulates the adapter with alarms after reestablishing the connection. Enter –1 to disable OvORB connection checking. The default is 15 seconds.
ORB Port	The TCP/IP port number where the Operations Center OvORB listens. The default is 1572. It is possible to integrate the HP OpenView (NNM) adapter and Operations Center without the OvORB. Leave the ORB port blank (the default). Start the NNM adapter and it should connect to the ovw map; the adapter icon should change to green. If the connection is unsuccessful, the icon is red (CRITICAL). For more information, see “Port Communications Setup” on page 91.
ORB Reconnect Delay	When the OvORB connection is down, attempts a reconnection to the OvORB using the specified number of seconds. Enter –1 to disable OvORB reconnection attempts. The default is 10 seconds.
OVW Connection Check Delay	Checks the connection to the OVW at the specified time interval (in seconds). If the connection is down, all alarms are removed from the adapter and elements change to the UNKNOWN state. Alarms and element state are repopulated after reestablishing the connection. Enter –1 to disable OVW connection checking. The default is 15.
OVW Connection Wait	The number of seconds to wait for a connection attempt to an OVW session. The default is 60.
OVwDB Port	The HPOpenView database port number where the Operations Center adapter connects. The default is 2447. For more information, see “Port Communications Setup” on page 91.
OperationPort	The port used by the OvORB to communicate with Operations Center for ping/traceroute operations. The default is 1572. For more information, see “Port Communications Setup” on page 91.
OvSNMP Filter	A filter used to determine the events forwarded from OpenView to the OvORB. The default is `{ALL}` .*
Script.onError	A script that executes if the adapter fails for any reason. For example, the script can print the reason for the failure as msg using log.info(msg).
Script.onInitialized	A script that executes when the adapter initializes. All of the `Script.*` properties are optional.
Script.onStarted	A script that executes when the adapter starts, either manually or automatically when the Operations Center server starts.
Script.onStopped	A script that executes after manually stopping the adapter.
Server Host	The hostname of the server where the management software is installed.

Property	Specify…
Session ID	The session ID to use to connect to NNM. Specify a value using one of the following formats: ◆ **n:** Connects to session *n*. ◆ **n…m:** Tries successively starting at session *n* and running to session *m*. Append with `ro` to connect only if the map is read/only or append with `rw` to connect only if the map is read/write. For example: Blank — Uses session instance 0 2 — Uses session instance 2 2…4 — Searches for a running instance from 2 to 4 2…6ro — Searches for a running read/only map from 2 to 6 2…8rw — Searches for a running read/write map from session 2 to 8 …10 rw — Searches for a running read/write map from 0 to 10
Severity Mapping	Maps one or more OpenView Alarm severity codes to Operations Center alarm severity codes. The default is blank, which uses the default mappings. To define a particular code mapping, use the following format, separating each mapping with a semicolon: `Openview=Formula;Openview=Formula;` Table A-14 shows the default mappings between OpenView and Operations Center, and Table A-15 lists possible Operations Center code mapping values. Value comparisons are case sensitive, so Critical is different from critical.
Stylesheet File	The stylesheet file in the */OperationsCenter_install_path/*`database` directory that applies to the HierarchyFile as a style markup and produces the final output.
Use Alarm Times For Condition Changes	The date/time stamp to use for all alarm data stored by the Operations Center Data Warehouse. If `true`, the alarm's date/time stamp is used. If `false`, the date/time stamp of when the Operations Center server received the alarm is used. The default is `true`. Alarm history is stored based on the alarm time rather than alarm receipt time. Also, for SLA metric data based on alarm properties, the property values are recorded based on the alarm time instead of the alarm receipt time. Note that recording historical condition data for historical alarms is not supported.
XY Layout Ratio	The spacing between NNM icons in the *Layout* view. Increase the value to increase the amount of space between icons. The default is 1.5.

The following sections provide reference information regarding Element Condition and Alarm Severity mappings for OpenView which can be updated by changing the `Condition Mapping` and `Severity Mapping` adapter properties.

- Section A.13.1, "Element Condition Mappings," on page 305
- Section A.13.2, "Alarm Severity Mappings," on page 305

A.13.1 Element Condition Mappings

Table A-14 lists the default mapping for OpenView states. Use the Condition Mapping adapter property to change any of the mappings. The mappings are case sensitive.

Table A-14 *Mappings for OpenView State*

OpenView State	Shows as:
None	UNMANAGED
Unknown	UNKNOWN
Normal	OK
Minor	MINOR
Critical	CRITICAL
Unmanaged	UNMANAGED
Warning	INFO
Major	MAJOR
Restricted	INFO
Testing	UNKNOWN
Disabled	UNMANAGED
Up	OK
Marginal	MINOR
Down	CRITICAL

A.13.2 Alarm Severity Mappings

Table A-15 lists the default severity mappings for OpenView severities. Use the SeverityMapping adapter property to change the mappings. The mappings are case sensitive.

Table A-15 *OpenView Severity Mappings*

OpenView Severities	Shows as:
Normal	OK
Warning	INFO
Minor	MINOR
Major	MAJOR
Critical	CRITICAL

A.14 HP Network Node Manager i-series

Table A-16 *HP Network Node Manager i-series Adapter Properties*

Property	Specify…
Alarm Columns	A comma-separated list that determines which alarm columns display and the order in which the alarm items display (date/time, rule, etc.) in the *Alarms* view. The default list is: `lifecycleStateShort,priorityShort,assignedTo,categoryShort,message`
HTTP Protocol	Specify the protocol to connect to the NNMi web server for SOAP connections. The default is `http`.
Hierarchy File	A file in the */OperationsCenter_install_path/*`database` directory that contains an XML description of the element hierarchy to build below the adapter element. The default is `examples/HPNNMiHierarchy.xml`.
Max Records Per Query	During a poll, NNMi information is retrieved in batches of records, until all records have been processed. Specify the number per query request to process before the next SOAP query. Set this property to limit the number of results returned per SOAP query alleviate issues with result size. For example, if a million records get returned, perhaps the NNMi web server has an issue. If so, can throttle down max records per query. Default is `500`.
Mine Closed Incidents Time	Controls the initial mining of alarms with a Closed status when the adapter starts. Default is 0; closed incidents are not mined, only Open and Acknowledged alarms are mined.
NNMi Host	The name of the NNMi web server host.
NNMi Password	The password for the NNMi web server host.
NNMi Port	The port for the NNMi web server host.
NNMi Username	The user name for the NNMi web server host.
Poll Period (secs)	The number of seconds to wait before starting a new poll. Default is 60.
Process Inventory Alarms	Specify if the Inventory alarms for Node Groups, Nodes, Interfaces, IP Addresses, IP Subnets, and L2 Connections are sent through the hierarchy file for hierarchy file processing. Default is `true`. If set to `false`, you can remove the Inventory group from the hierarchy file.
Script.onError	A script that executes if the adapter fails for any reason. For example, the script can print the reason for the failure as msg using log.info(msg).
Script.onInitialized	A script that executes when the adapter initializes.
Script.onStarted	A script that executes when the adapter starts.
Script.onStopped	A script that executes after manually stopping the adapter.
Stylesheet File	The stylesheet file in the */OperationsCenter_install_path/*`database` directory that applies to the HierarchyFile as a style markup to produce the final output.
Topology Inventory Folder Name	The label to use for the *Topology Inventory* folder in the elements tree.

Property	Specify…
Topology Maps Folder Name	The label to use for the *Topology Maps* folder in the elements tree.
Use Alarm Times For Condition Changes	The date/time stamp to use for all alarm data stored by the Operations Center Data Warehouse. If `true`, the alarm's date/time stamp is used. If `false`, the date/time stamp of when the Operations Center server received the alarm is used. The default is `true`. Alarm history is stored based on the alarm time rather than alarm receipt time. Also, for SLA metric data based on alarm properties, the property values are recorded based on the alarm time instead of the alarm receipt time. Note that recording historical condition data for historical alarms is not supported.
Use Extended API	Indicate if NNMi queries for structure information regarding Nodes and Node Groups. I.e. children of a Node Group, so can reflect that structure in our element tree. Default is `true`.
Use NNM Icons for Nodes	If set to true and if *Use Extended API* is set to true, then NNMi is queried for the Node icons to use in the element tree and layout view. Default is `true`. If `false`, standard icons are used.
Use Pattern for IP Address Inventory	The integration discovers inventory objects from NNMi (Nodes, Node Groups, Interfaces, IP Addresses, IP Subnets, L2 Connections). Objects are placed under the *Topology Inventory* element tree branch. Specify patterns to create subfolders under that branch for the Inventory items. For example, if there are thousands of IP Addresses and no pattern is set, they all show under the *Topology Inventory/IP Addresses* element tree branch as peers. Using a pattern, the IP Addresses can be broken into subfolders based on portions of the ip address name. For IP Addresses, the default pattern is based on the first three numbers of the IP Address, so IP Address 192.72.13.200 would be placed in the element tree under parent *Topology Inventory/IP Addresses/192.72.13*. Specify the pattern: ◆ Start the pattern with a minus sign to disable. Inventory items will be a flat list in the elements tree. ◆ Use one or more regular expressions separated by commas. Surround the portion of the regular expression that should become the folder name with parentheses. For example, assume this is the pattern for interfaces: `(..).*,(....).*,(……).*` This pattern would create at most three sub folders to contain the interfaces. The subfolders would be based on the first two characters, then the first four characters, then the first six characters of the Interface name. So interface "`ethernet64` would be placed in *Topology Inventory/Interfaces/et/ether/ethern* or three subfolders deep since the name is longer than six characters. Interface `lo5` would be placed in *Topology Inventory/Interfaces/lo* or one subfolder deep since the name is between two and four characters.
Use Pattern for IP Subnet Inventory	Same as the *Use Pattern for IP Address Inventory* property above but for IP Subnets. No default pattern.
Use Pattern for Interface Inventory	Same as the *Use Pattern for IP Address Inventory* property above but for Interfaces. No default pattern.

Property	Specify…
Use Pattern for L2 Connection Inventory	Same as the *Use Pattern for IP Address Inventory* property above but for L2 Connections. No default pattern.
Use Pattern for Node Group Inventory	Same as the *Use Pattern for IP Address Inventory* property above but for Node Groups. No default pattern.
Use Pattern for Node Inventory	Same as the *Use Pattern for IP Address Inventory* property above but for Nodes. No default pattern.

A.15 HP OpenView Operations for UNIX

Table A-17 *HP OpenView Operations for UNIX Adapter Properties*

Property	Specify…
Alarm Columns	A comma-separated list that determines which alarm columns display and the order in which the alarm items display (date/time, rule, etc.) in the *Alarms* view. The default list is: `Acknowledged,Owned,Application,Object,Message Group,Message Text`
DB Host	The name of the host on which Oracle runs. Oracle and OpenView can run on separate servers. If this property value is blank, the value entered for the ServerHostname property is used. This is a required property.
DB Name	The database name. This is a required property. The default is `openview`.
DB Password	The password of the Oracle database user.
DB Poll	The number of seconds between queries for new alarms or re-queries for updating existing alarms. This is a required property. The default is 30.
DB Port	The database port. The default is 1521. This is a required property.
DB Type	The database type. This is a required property. The only valid value is oracle.
DB User	The Oracle database user name. Use the opc_op or opc_report created when installing Operations. This is a required property.
Elements Timeout	The length of time, in seconds, to retain alarm elements. If no open alarms exist and the element's condition does not change in n seconds, and the element has no children, then the element disappears. The element redisplays if another alarm is generated. The default is 300 seconds. **AgeOutTime < 0:** Never age out. **AgeOutTime = 0:** Age out immediately. **AgeOutTime > 0:** Age out after specified time expires.
Enforce Responsibility Matrix	If True, the integration filters alarms based on the user's responsibility matrix as configured in HP OpenView Operations. If False, no alarms are filtered. The default is True.
Hierarchy File	A file in the */OperationsCenter_install_path/*`database` directory that contains an XML description of the element hierarchy to build below the adapter element. The default is `examples/ITOHierarchy.xml`.

Property	Specify…
History Alarm Retrieval Timeperiod	The number of days to poll and retrieve in event history; active at adapter startup. Enter 0 to disable retrieval. Enter -1 to retrieve the entire available history. The default is 7.
Max Alarms	The maximum number of alarms that the adapter queries and retains. The default is 50000. Enter 0 to allow an unlimited number of alarms.
OVO Password	The password of the user specified in the OVO User property.
OVO Server Host	The server on which the Operations management system runs. If a location is specified, the integration is bidirectional.
OVO Server Port	The port number for the OVO Server Host. The default is 1578.
OVO User	The user name that allows Operations Center to access the Operations management system. This user name can be the same as that of the administrator. This is a required property.
OwnedAffectsCondition	If True, alarms marked as “Owned” do not propagate state within Operations Center. If False, Owned alarms do propagate state.
Script.onError	A script that executes if the adapter fails for any reason. For example, the script can print the reason for the failure as msg using log.info(msg).
Script.onInitialized	A script that executes when the adapter initializes.
Script.onStarted	A script that executes when the adapter starts.
Script.onStopped	A script that executes after manually stopping the adapter.
Stylesheet File	The stylesheet file in the */OperationsCenter_install_path/*`database` directory that applies to the HierarchyFile as a style markup to produce the final output.
Use Alarm Times For Condition Changes	The date/time stamp to use for all alarm data stored by the Operations Center Data Warehouse. If `true`, the alarm’s date/time stamp is used. If `false`, the date/time stamp of when the Operations Center server received the alarm is used. The default is `true`. Alarm history is stored based on the alarm time rather than alarm receipt time. Also, for SLA metric data based on alarm properties, the property values are recorded based on the alarm time instead of the alarm receipt time. Note that recording historical condition data for historical alarms is not supported.

A.16 HP ServiceCenter and HP Service Manager

Table A-18 *HP ServiceCenter and HP Service Manager Adapter Properties*

Property	Specify…
Alarm Columns	A comma-separated list that determines which alarm columns display and the order in which the alarm items display (source of alarm, alarm class, etc.) in the Alarms view. The default is: `Key=_SC_RECORD_KEY, Status=_SC_STATUS, Assigned To=_SC_ASSIGNED_TO, Module=_SC_MODULE_NAME, Description=_SC_DESCRIPTION`.

Property	Specify…
Configuration File	A relative file name in the */OperationsCenter_install_path/*database directory. The file contains ServiceCenter configuration settings. The default is examples/ServiceCenterConfiguration.xml or examples/ ServiceManagerConfiguration.xml. If integrating to HP Service Manager 9.3, reference examples/ ServiceCenterConfiguration_9.3.xml, or copy and modify to customize.
Hierarchy File	A relative file name in the */OperationsCenter_install_path/*database directory. The file contains an XML description of the hierarchy of elements that is built below the element that represents the adapter. The default is examples/ ServiceCenterHierarchy.xml or examples/ ServiceManagerHierarchy.xml.
Max Alarms Per Module	The maximum number of alarms that Operations Center actively maintains per ServiceCenter or Service Manager Module. Exceeding this number removes the oldest alarm. The default is 500.
Polling Interval	The number of seconds between queries for new tickets or re-queries for updating existing tickets. The default is 60.
Script.onError	A script that executes if the adapter fails for any reason. For example, the script can print the reason for the failure as msg using log.info(msg).
Script.onInitialized	A script that executes when the adapter initializes. All of the Script.* properties are optional.
Script.onStarted	A script that executes when the adapter starts, either manually or automatically when the Operations Center server starts.
Script.onStopped	A script that executes after manually stopping the adapter.
ServiceCenter/Service Manager Host Name	The name of the host on which ServiceCenter or Service Manager Soap server resides. This is a required property.
ServiceCenter/Service Manager Integration API Timezone	Specify the time zone for the events sent by the ServiceCenter or Service Manager Integration API. The default is GMT. To specify a different time zone, use the format: GMT+/–HH:MM, where hours and minutes are added or subtracted from GMT. For example, Eastern Standard Time (EST) is: GMT-05:00. Common time zone abbreviations such as EST or PST (Pacific Standard Time) are also acceptable. This property is NOT the same as the time zone of the ServiceCenter server. The ServiceCenter server itemizing depends on the geographical location of the server.
ServiceCenter/Service Manager Password	The corresponding password for the provided user.
ServiceCenter/Service Manager Port	The port number on which the ServiceCenter or Service Manager Soap server listens. The exception is if you run ServiceCenter or Service Manager manually and use a different port. Then the port number is usually found in the ServiceCenter or Service Manager RUN\sc.ini file. For example, an entry of system:12670 in the sc.ini file requires using 12670 for this adapter property. This is a required property.
ServiceCenter/Service Manager User Name	The user name for the ServiceCenter or Service Manager user account. This user name must have sufficient ServiceCenter or Service Manager authorization to perform all defined Alarm and Element operations. This is a required property.
Show Query Info	If True, displays query statistics in the Operations Center log per schema, per poll period, after creating, updating, or deleting an alarm. The default is False.

Property	Specify…
Stylesheet File	The stylesheet file in the */OperationsCenter_install_path/*`database` directory that applies to the HierarchyFile as a style markup to produce the final output.
Use Alarm Times For Condition Changes	The date/time stamp to use for all alarm data stored by the Operations Center Data Warehouse. If `true`, the alarm's date/time stamp is used. If `false`, the date/time stamp of when the Operations Center server received the alarm is used. The default is `true`. Alarm history is stored based on the alarm time rather than alarm receipt time. Also, for SLA metric data based on alarm properties, the property values are recorded based on the alarm time instead of the alarm receipt time. Note that recording historical condition data for historical alarms is not supported.

A.17 IBM Micromuse Netcool

Table A-19 *IBM Micromuse Netcool Adapter Properties*

Property	Specify…
AccountMap	The file used to construct a map between Operations Center and Netcool users. The comments at the top of this file provide details on how to assemble the AccountMap file. The default is `examples/NetcoolAccountMap.properties`.
AckAffectsCondition	If True, an acknowledged alarm contributes to element condition and alarm counts. The default is True. For example, assume there is an element with one critical event: `counts= 1-CRITICAL element=RED` Acknowledge the alarm and set AckAffectsCondition to True. This results in reducing the alarm count by one and changing the element condition to green: `counts= 0-CRITICAL element=GREEN` Acknowledge the alarm and set AckAffectsCondition to False. The result is the alarm still exists and the condition remains unchanged after acknowledging it: `counts =1-CRITICAL element=RED`

Property	Specify…
AlarmColumnDefinition	By default, the alarm property pages display all original Netcool fields in the same order used on the Netcool Objectserver. To change the properties or order in which they display, edit the blank, default file named */OperationsCenter_install_path*/database/examples/NetcoolAlarmColumns.properties. Then specify the path and file name in this property. Edit the file to specify the order in which properties display or to exclude properties from display: ◆ To set the order of properties displayed in the alarm property pages, enter the property names one per line in the preferred order. ◆ To exclude a field, enter the field name followed by an equal sign and the word exclude. In the following example, the *Serial* field displays at the top of the Properties page and *NodeAlias* does not display in the property page: `Serial NodeAlias=exclude` If no file or an empty file is specified, all alarm columns display in the same order used on the Netcool Objectserver.
AlarmColumns	A comma-separated list that determines which alarm columns display and the order in which they display in the *Alarms* view. To conserve memory and alarm history storage space, specify only the relevant alarm columns and exclude mining of all other alarm data. There are several ways to specify alarm columns: ◆ Use the format *colum_nname*=included to include an alarm column; for example: Customer=included. Shortcut: Identify only “included” fields and all other fields are excluded. ◆ Use the format *column_name*=excluded to exclude an alarm column; for example: Customer=excluded. Shortcut: Identify only “excluded” fields and all other fields are included. ◆ If an alarm column is identified as both included and excluded, then it is excluded. ◆ If only alarm column names are listed without any “included” or “excluded” tags, then the listed columns display in the *Alarms* view, but all other alarm data is mined. ◆ The fields that are always included: Serial, Severity, LastOccurrence, Acknowledged, OwnerUID, OwnerGID, and Summary.
AlarmSeverityMapping	A comma delimited list of severity mappings for alarms. ◆ Use the format *NetcoolSeverity=OperationsCenterSeverity* For example: CLEAR=OK,INDETERMINATE=INFORMATIONAL,WARNING=MINOR,MINOR=MINOR,MAJOR=MAJOR,CRITICAL=CRITICAL
DatabaseDriver	Version of Sybase JDBC drivers being used by Operations Center. The default is Sybase. For example, to use Sybase version 3 drivers, save the drivers to the */OperationsCenter_install_path*/classes/ext directory, then update the DatabaseDriver adapter property to com.sybase.jdbc3.jdbc.SybDriver.

Property	Specify…
ElementsTimeout	If there are no open alarms and the element's condition has not changed in the last n seconds, and the element has no children, then the element disappears from the display. If another alarm is generated for this element, then it reappears. The default is 300. **ElementsTimeout < 0:** Never time out. **ElementsTimeout = 0:** Time out immediately. **ElementsTimeout > 0:** Time out after specified time expires.
HierarchyFile	A file in the */OperationsCenter_install_path/*`database` directory that contains an XML description of the element hierarchy to build below the adapter element. The default is `examples/NetcoolHierarchy.xml`.
HostTokens	A list of Netcool token values that determine console connectivity. If Operations Center finds one of these tokens in an alarm, it uses the token to assign connectivity. The default is `Node`.
IDUC Timer	The time interval used to check for updates for a specific Netcool/Operations CenterOperations Center integration. Overrides the Netcool global IDUC timer setting. The default is 60 seconds. Valid values are greater than 5 seconds.
MaxAlarms	The maximum number of alarms that the adapter queries and retains. The default is 0, which allows an unlimited number of alarms.
ObjectServerAccount	The user name for connecting to the object server. The default is `root`.
ObjectServerHost	The host location of the object server.
ObjectServerName	The name of the object server. The default is `NCOMS`.
ObjectServerPassword	The password for the user name supplied as the ObjectServerAccount.
ObjectServerPort	The port on which the object server host listens. The default is 4100.
ObjectServerVersion	The version of Object Server software that is installed. The default is `7.4.0`.
OperationsMenu	The Netcool object server menu table used to populate the right-click operations for Netcool alarms. If blank, the default *AlertsMenu Netcool* menu displays. If the adapter attempts but fails to use the specified menu table, no operations display and the `formula.trc` file logs a warning.
RelayServer	The name of the server on which the relay connection exists. For details on setting up a relay connection, see "Integration Using a Secure Relay Connection" on page 107.
RelayServerPort	The port number configured for use by the Netcool adapter for relay communications.
RelaySecurity	The security level for the relay server: SSL or unsecured (meaning use cleartext, which is not case-sensitive).

Property	Specify…
Script.onConnected	A script that executes when the adapter successfully connects to the object server. The default is: `@adapters/Netcool/setServerSkewTime.fs` This script calculates the time differential (skew) between the Operations Center and object servers. This skew affects calculations of the date/time stamps used for user presentation and information logging/stamping. If the Operations Center server runs on the same machine as the Netcool server, then there is no skew to calculate. In this case, the time skew script is unnecessary. If a site synchronizes the time for its servers using the standard UNIX timeserver service (usually port 37), then the difference in time between the Operations Center server and the machine running the Netcool server might be negligible. In this case, ignore this parameter. The supplied script calculates the difference in time between the Operations Center server and the target Netcool server by running the REXEC protocol to invoke the `date` command on the UNIX host. If the Netcool server runs another operating system, or if a site has restricted the access to the REXEC protocol, use a different mechanism to calculate the difference in time between the machine running Operations Center and the Netcool server. If using a different mechanism, study the supplied script as a model for how to make the adjustments to the Operations Center Netcool adapter. The script invokes the `setServerTimeSkew( )` adapter method using the number of milliseconds (seconds times 1000) of difference between the Operations Center server and Netcool server.
Script.onDisconnected	A script that executes when the adapter disconnects from the object server. All of the `Script.*` properties are optional. A script that executes if the adapter fails for any reason. For example, the script can print the reason for the failure as msg using log.info(msg).
Script.onError	A script that executes if the adapter fails for any reason. For example, the script can print the reason for the failure as msg using log.ino(msg).
Script.onInitialized	A script that executes when the adapter initializes.
Script.onStarted	A script that executes when the adapter starts.
Script.onStopped	A script that executes after manually stopping the adapter.
StylesheetFile	The stylesheet file in the */OperationsCenter_install_path/*`database` directory that applies to the HierarchyFile as a style markup to produce the final output.
UnacknowledgedText	The text string to match that indicates that the alarm is unacknowledged. Defaults to `no`.
UseAlarmTimesForCond Changes	The date/time stamp to use for all alarm data stored by the Operations Center Data Warehouse. If `true`, the alarm's date/time stamp is used. If `false`, the date/time stamp of when the Operations Center server received the alarm is used. The default is `true`.

A.18 IBM Tivoli Application Dependency Discovery Manager (TADDM)

Table A-20 *IBM Tivoli Application Dependency Discovery Manager Adapter Properties*

Property	Specify…
AlarmColumns	A comma-separated list that determines which alarm columns display and the order in which the alarm items display in the *Alarms* view. The defaults are: `Status,Class,Description`. Aliases for Alarm Columns: Assign new names to alarm columns using the format: *`display_name=current_name`*. For example, `Condition=Status` displays status data in a column named Condition.
HierarchyFile	A file in the */`OperationsCenter_install_path`/`database` directory that contains an XML description of the element hierarchy to build below the adapter element. The default is `/examples/TADDMHierarchy.xml`.
Script.onError	A script that executes if the adapter fails for any reason. For example, the script can print the reason for the failure as msg using log.info(msg).
Script.onInitialized	A script that executes when the adapter initializes.
Script.onStarted	A script that executes when the adapter starts, either manually or automatically, when the Operations Center server starts.
Script.onStopped	A script that executes after manually stopping the adapter.
StyleSheet File	The stylesheet file in the */`OperationsCenter_install_path`/`database` directory that applies to the HierarchyFile as a style markup to produce the final output.
TADDM Host	The TADDM server hostname.
TADDM Password	The password associated with the user name for logging into the TADDM server.
TADDM Port (API)	The API port number for the TADDM server. The TADDM server might have a `collation.properties` configuration file that contains the API port number.
TADDM Port (RMI)	The RMI port number for the TADDM server. The RMI port can be found by using the Web client and clicking *Start Product Client* to launch the Java client. The RMI port is listed at the bottom of the user login page.
TADDM Refresh Interval (minutes)	**(IBM Tivoli Application Dependency Discovery Manager 7.x adapter only)** The poll interval, in minutes, for looking for new discovered data. Set to `15` minutes or greater. Set to `0` for no polling. Default is 60. For more information about using a refresh interval, see “Updating TADDM Data” on page 152.
TADDM Username	The user name for logging into the TADDM server.

Property	Specify…
Use Alarm Times For Condition Changes	The date/time stamp to use for all alarm data stored by the Operations Center Data Warehouse. If `true`, the alarm's date/time stamp is used. If `false`, the date/time stamp of when the Operations Center server received the alarm is used. The default is `true`. Alarm history is stored based on the alarm time rather than alarm receipt time. Also, for SLA metric data based on alarm properties, the property values are recorded based on the alarm time instead of the alarm receipt time. Note that recording historical condition data for historical alarms is not supported.

A.19 IBM Tivoli NetView

Table A-21 *IBM Tivoli NetView Adapter Properties*

Property	Specify…
Command.Ping	The command used for the `Ping` function. The syntax varies among operating systems. The default is `ping -t`.
Command.Trace.Route	The command used for the `TraceRoute` function. The syntax varies among operating systems. The default is `tracert`.
ConditionMapping	Maps one or more NetView conditions to Operations Center severity codes. The default is blank, which uses the default mappings. To define a particular code map, use the following format, separating each mapping with a semicolon: `Netview_condition=Formula_code;Netview_condition=Formula_code;` Table A-22 shows the default mappings between NetView and Operations Center.
DiscoveryDepth	The number of levels that the adapter loads upon startup from the management software. The default is 2.
MaxAlarms	The maximum number of alarms that the adapter queries and retains. Exceeding this value removes alarms on a first in, first out basis. Enter 0 to allow an unlimited number of alarms. The default is 500.
NetViewCharacterEncoding	For use with some non-English character sets. Performs character conversion when set to a canonical character set name, as listed at: (http://java.sun.com/j2se/1.3/docs/guide/intl/encoding.doc.html).
RequestDepth	The number of levels that the adapter opens from the management software after opening a branch using the Operations Center console. The adapter performs an incremental load, which prevents overloading the system when large managed networks are involved. The default is 2. This setting affects elements below the second level of the element hierarchy only. Requests made for the root element (1) returns only the first two levels.
Script.onConnected	A script that executes when the adapter successfully connects to the object server. All of the `Script.*` properties are optional.
Script.onDisconnected	A script that executes when the adapter disconnects from the object server.

Property	Specify…
Script.onError	A script that executes if the adapter fails for any reason. For example, the script can print the reason for the failure as msg using log.info(msg).
Script.onInitialized	A script that executes when the adapter initializes.
Script.onStarted	A script that executes when the adapter starts, either manually or automatically when the Operations Center server starts.
Script.onStopped	A script that executes after manually stopping the adapter.
ServerHost	The hostname of the server where the management software is installed.
ServerPort	The TCP/IP port number on which the Operations Center ORB listens. The default for NetView and OpenView is 1572.
SeverityMapping	Maps one or more NetView severity codes to Operations Center severity codes. The default is blank, which uses the default mappings. To define a particular code map, use the following format, separating each mapping with a semicolon: `Netview_code=Formula_code;Netview_code=Formula_code;` For example, to change the severity mapping of indeterminate alarms within NetView to UNMANAGED within Operations Center, change the SeverityMapping value to: `indeterm=unmanaged;` Table A-22 shows the default mappings between NetView and Operations Center, and Table A-23 lists the possible Operations Center code mapping values.

Table A-22 lists the default mapping between NetView states and Operations Center codes. Use the ConditionMapping adapter property to change any of the mappings.

Table A-22 *NetView State and Operations Center Default Mapping*

NetView State Values	Operations Center Codes
normal	OK
acknowledge	OK
minor	MINOR
critical	CRITICAL
unmanaged	UNMANAGED
major	MAJOR
restricted	INFORMATIONAL
testing	INFORMATIONAL
unknown	UNKNOWN

Table A-23 lists the default severity mappings between NetView and Operations Center. Use the SeverityMapping adapter property to change any of the mappings.

Table A-23 *Default NetView and Operations Center Severity Code Mappings*

NetView Severity Codes	Operations Center Severity Codes
cleared	OK
unknown	UNKNOWN
indeterm	UNKNOWN
warning	INFORMATIONAL
minor	MINOR
major	MAJOR
critical	CRITICAL

The following list consists of valid Operations Center values for state/severity mapping:

- UNKNOWN
- CRITICAL
- MAJOR
- MINOR
- INFO
- INFORMATIONAL
- OK
- INITIAL
- UNMANAGED
- USAGE_IDLE
- IDLE
- USAGE_ACTIVE
- ACTIVE
- USAGE_BUSY
- BUSY

Some values are synonyms of others. For example, INFO and INFORMATIONAL are the same, as are usage_idle and idle. The UNMANAGED and INITIAL values are also synonyms.

A.20 IBM Tivoli Enterprise Console (T/EC)

When entering T/EC adapter property values, include two single quotes instead of a single quote within a single-quoted string. For example, a correct entry for adding a single quote before and after the word single:

```
msg='double ''single'' quotes'
```

Table A-24 *IBM Tivoli T/EC Adapter Properties*

Property	Specify…
AckAffectsCondition	If True, an acknowledged alarm contributes to element condition and alarm counts. The default is True. For example, assume there is an element with one critical event: `counts= 1-CRITICAL element=RED` Acknowledge the alarm and set AckAffectsCondition to True. This results in reducing the alarm count by one and changing the element condition to green: `counts= 0-CRITICAL element=GREEN` Acknowledge the alarm and set AckAffectsCondition to False. The result is the alarm still exists and the condition remains unchanged after acknowledging it: `counts= 1-CRITICAL element=RED`
AcknowledgeAvailable	If True, the *Acknowledge* option is available across the entire instance of the T/EC adapter. If False, the *Acknowledge* option is not available on the alarm right-click menu, regardless of the user access privileges. The default is True.
AlarmColumns	A comma-separated list that determines which alarm columns display and the order in which the alarm items display in the *Alarms* view. The defaults are: `Status`,`Class`,`Description`. Aliases for Alarm Columns: Assign new names to alarm columns using the format: *display_name*=*current_name*. For example, `Condition=Status` displays status data in a column named *Condition*.
CloseAvailable	If True, the *Close* option is available across the entire instance of the T/EC adapter. If False, the *Close* option is not available on the alarm right-click menu, regardless of the user access privileges. The default is True.
ClosedAlarmsTimeout	The length of time, in seconds, to display an alarm after closing it in the Operations Center console. Enter -1 to display the alarm indefinitely. Enter 0 to remove it immediately. The default is 1800.
ElementsTimeout	If there are no open alarms, and the element's condition hasn't changed in the last n seconds, and the element has no children, then the element disappears from the display. If another alarm is generated for this element, then it reappears. The default is 300 seconds. **ElementsTimeout < 0:** Never time out. **ElementsTimeout = 0:** Time out immediately. **ElementsTimeout > 0:** Time out after specified time expires.
EventConsoleName	The ID used by the Operations Center server to identify itself to T/EC. This value must be a valid T/EC Enterprise client name of the T/EC system being integrated. For T/EC 3.6, to look up console names, use the `wlookup –ar Enterprise Client` command from the T/EC server or IBM Tivoli TMR. The default is `@Formula`.
EventListenPort	The TCP/IP socket port number for which T/EC is configured to forward its events. Use any port number (above 1000 in UNIX). The default is 12345.

Property	Specify…
ForceDateUpdate	If True, overrides the alarm date with the T/EC date_reception or date slot if available. If set to True and UseAlarmTimesForConditionChanges is True, historical alarms can be imported for use with BSLM. Condition data generates from historical alarms and key performance metrics can be stored using the historical alarm properties. The default is False.
HierarchyFile	A file in the */OperationsCenter_install_path/*`database` directory that contains an XML description of the element hierarchy to build below the adapter element. The default is `examples/TecHierarchy.xml`.
HostsToMine	Use one of the following values: ◆ **Blank:** This property only affects the mining of open/acknowledged events when the adapter starts. The adapter functions properly without any entry. The difference is if it discovers all previously opened/acknowledged events or only new events as they occur. ◆ **List of hostnames:** A T/EC adapter can connect and multiplex many ORBs on many hosts. This entry can consist of a list of hostnames separated by commas. Each ORB can listen on a different port. This entry can contain a more complex entry such as: `hostname:ORBPort` A T/EC server can listen on different ports. This entry can contain a more complex entry such as: `hostname:ORBPort:TECPort` Assume the following configuration: 1) host x, ORB listening on port 9990, T/EC Server listening on port 9991 2) host y, ORB listening on port 9995, T/EC Server listening on port 9996 Set the adapter property HostsToMine to: `x:9990:9991,y:9995:9996`
IdDateMethod	If True, Operations Center generates the alarm date based on the T/EC originating_event_id alarm column. If False, Operations Center generates the alarm date based on the T/EC date_reception alarm column. The default is False.
MaxActiveReaders	The maximum number of threads reading at one time. The default is 1.
MaxAlarms	The maximum number of alarms that the adapter queries and retains. Exceeding this number removes the oldest alarm. Enter 0 to allow an unlimited number of alarms. The default is 500.
MiningLimit	The maximum number of T/EC events that the ORB retrieves from each T/EC host. The default is 2000.
Script.onError	A script that executes if the adapter fails for any reason. For example, the script can print the reason for the failure as msg using log.info(msg).
Script.onInitialized	A script that executes when the adapter initializes. All of the `Script.*` properties are optional.
Script.onStarted	A script that executes when the adapter starts, either manually or automatically when the Operations Center server starts.

Property	Specify…
Script.onStopped	A script that executes after stopping the adapter.
SeedFile	A file in the T/EC reception log format that contains information about its managed elements. This file enables the Operations Center server to generate the elements hierarchy displayed in the Operations Center console.
SeverityMapping	Maps the T/EC severity codes to Operations Center severity codes. List the T/EC severity code first. The default is: `Fatal=Critical;Critical=Critical;Minor=Minor;Warning=Major;Harmless=Informational;Unknown=Unknown`
StylesheetFile	The stylesheet file in the */OperationsCenter_install_path/*`database` directory that applies to the HierarchyFile as a style markup to produce the final output.
SuppressAvailable	If True, the Suppress Alarm operation is available across the entire instance of the T/EC adapter. If False, the *Suppress Alarm* right-click option is not available regardless of ACLs. The default is True.
Suppression Time	The time in seconds that an alarm can remain in the suppressed state. Enter 0 to disable the feature. The default is 1800. The T/EC adapter Suppress Alarm alarm operation suppresses any alarm with an OPEN status for a fixed amount of time. The severity level for a suppressed alarm changes to SUPPRESSED and the alarms severity does not populate up the element hierarchy. The alarm remains in this state until one of the following occurs: ◆ The operator closes the alarm ◆ The alarm changes severity to OK ◆ The timer expires ◆ The operator uses the Unsuppress Alarm operation
SyncClass	The default is `TEC_sync`, which allows multiple instances of T/EC to synchronize their alarms.
TecORBPort	The TCP/IP port number on which the Operations Center ORB listens. The TecORB default is 1576.
UseAlarmTimesFor CondChanges	The date/time stamp to use for all alarm data stored by the Operations Center Data Warehouse. If `true`, the alarm's date/time stamp is used. If `false`, the date/time stamp of when the Operations Center server received the alarm is used. The default is `true`.
WTDumperCommand	If using ORB Integration, the ORB issues this command upon startup for initial data mining, to pipe events back. The default is: `wtdumper -o DESC -dw "status<='20' AND severity>='20'"` Edit the command based on the database syntax requirements.

A.21 IBM Tivoli Enterprise Console (T/EC)+, Database Edition

Table A-25 *IBM Tivoli Enterprise Console (T/EC)+, Database Edition Adapter Properties*

Property	Specify…
AckAffectsCondition	If True, an acknowledged alarm contributes to element condition and alarm counts. The default is True. Assume there is one element with one critical event: `counts= 1-CRITICAL element=RED` Acknowledge the alarm and set AckAffectsCondition to True. This results in reducing the alarm count by one and changing the element condition to green: `counts= 0-CRITICAL element=GREEN` Acknowledge the alarm and set AckAffectsCondition to False. The result is the acknowledged alarm still exists: `counts= 1-CRITICAL element=RED` In the example above, we assumed there was only one alarm. In reality, any other existing alarms continue to contribute to the element's condition.
AlarmColumns	A comma-separated list that determines which alarm columns display and the order in which the alarm items display in the *Alarms* view. The defaults are: `-ID`, `T/EC ID=originating_event_id`, `Status`, `Administrator`, and `Description`. Aliases for Alarm Columns: Assign new names to alarm columns using the format: *`display_name=current_name`*. For example, `Condition=Status` displays status data in a column named Condition.
ClosedAlarmsTimeout	The length of time (in seconds) to display an alarm after closing it in the Operations Center console. Enter -1 to display the alarm indefinitely. Enter 0 to remove it immediately. The default is 1800. When storing historical data for T/EC events, set ClosedAlarmsTimeout to a value greater than the poll interval time specified for the DB.time property. Otherwise, alarms do not remain open long enough for a polling cycle to pick up and store as historical data.
DB.backoff.time	The number of seconds to extend queries, in addition to the standard 15 second query cycle. Slow or very busy T/EC servers occasionally experience event processing that takes longer than 15 seconds. Symptoms of such a problem include one or more alarms that remain open in Operations Center after closing them in T/EC. Resolve the problem by increasing this property value from the default of 5 seconds. The initial recommended value is 30 seconds, then an increase to 60 seconds, if the problem persists. Contact Support (http://www.netiq.com/support/) if the problem persists.
DB.database	The logical name of the database or partition where the T/EC tables reside within the T/EC database. For Oracle, this is the system ID (SID) of the database. For other databases, this is the database name. The default is `tec`.
DB.host	The host machine of the IBM Tivoli Event Console Database.
DB.password	The password for the DB.user account name.

Property	Specify…
DB.port	The TCP/IP communications port of the database, if different from system defaults for the given database type. This can be blank if using the default port. Default ports for database types supported are: Oracle: 1521 Sybase: 4100 MSSQL: 1433 DB2: 6789
DB.time	The poll interval, after initial discovery, for updated and new alarms. The default is 15 seconds.
DB.type	Enter one of the following database names: ◆ **oracle7:** Oracle 7 only ◆ **oracleoci:** Oracle native "OCI" ◆ **sybase:** Sybase SQL Server ◆ **sybase_tli:** Sybase SQL Server using TLI ◆ **mssql:** Microsoft SQL Server ◆ **db2:** IBM DB2 Universal Database The default is `oracle`. For information on setting up the Oracle driver or DB2 driver, see Section A.21.1, “Setup for Native "OCI" Oracle Driver,” on page 325 and Section A.21.2, “Setup for DB2,” on page 326. To use a different driver or database that is not supported by the predefined types, set up Operations Center to communicate with the database through a special syntax for the DB.type parameter. The syntax is as follows: `driver | url | [time_select_query]` The driver is a standard Java class setting, as specified by the JDBC driver documentation for the unsupported driver. Specify the URL according to the documentation provided by the vendor. An example, using the DB2 app driver: `COM.ibm.db2.jdbc.app.DB2Driver|jdbc:db2:tec|select current date` The last parameter is optional. Operations Center adjusts for time differences between management systems by obtaining the current date during initialization of the integration with the management source. This query is optional. It returns the current date of the database server using the native syntax of the database.
DB.user	The user account used to log in to the database. The default is `tec`.

Property	Specify…
ElementsTimeout	If there are no open alarms, and the element's condition hasn't changed in the last n seconds, and the element has no children, then the element disappears from the display. If another alarm is generated for this element, then it reappears. The default is 300 seconds. **ElementsTimeout < 0:** Never time out. **ElementsTimeout = 0:** Time out immediately. **ElementsTimeout > 0:** Time out after specified time expires.
HierarchyFile	A file in the */OperationsCenter_install_path/*`database` directory that contains an XML description of the element hierarchy to build below the adapter element. The default is `examples/TecHierarchy.xml`.
IntegrationStrategy	One of three types of T/EC integration strategies. For further information about these types and their required properties, see the T/EC + adapter integration steps. Specify the type followed by the required properties (using name/value pairs) separated by commas: ◆ To set a direct connection to the database, specify: `DB`. The default is `DB`. ◆ To retrieve events directly from the database, but push alarm updates to T/EC via TecORB using a Java implementation of the T/EC wsetemsg utility, specify: `ORB,hostname=TecORB hostname,port=TecORB port,console=consoleName` ◆ To retrieve events directly from the database, but push alarm updates to T/EC via TecORB using a Java implementation of the T/EC postemsg utility, type: `POSTEMSG,hostname=T/EC Enterprise Server hostname,port=T/EC Enterprise Server port,syncClass=event class` (defined in the T/EC rule base which expects an alarm update from Operations Center).
MaxAlarms	The maximum number of alarms that the adapter queries and retains. Exceeding this number removes the oldest alarm. The default is 0, which allows an unlimited number of alarms.
MineClosedAlarmsTime	Controls the initial mining of alarms with a Closed status when the adapter starts. The default is 0, meaning Closed alarms are not mined. Only Open and Acknowledged alarms are mined. Set to a number n greater than zero to mine Closed alarms that existed within the past n in minutes, in addition to mining Open and Acknowledged alarms.
Operations	Defines the available operations for alarms. The default is: `ack,close,reopen,suppress,assign`
Script.onError	A script that executes if the adapter fails for any reason. For example, the script can print the reason for the failure as msg using log.info(msg).
Script.onInitialized	A script that executes when the adapter initializes. All of the `Script.*` properties are optional.
Script.onStarted	A script that executes when the adapter starts, either manually or automatically when the Operations Center server starts.

Property	Specify…
Script.onStopped	A script that executes after stopping the adapter.
SeedFile	A file in the T/EC reception log format that contains information about its managed elements. This file enables the Operations Center server to generate the elements hierarchy displayed in the Operations Center console.
SeverityMapping	Maps the T/EC severity codes to Operations Center severity codes. List the T/EC severity code first. The default is: `Fatal=Critical; Critical=Critical;Minor=Minor; Warning=Major; Harmless=Informational; Unknown=Unknown`
StylesheetFile	The stylesheet file in the */OperationsCenter_install_path/*`database` directory that applies to the HierarchyFile as a style markup to produce the final output.
Suppression Time	The time, in seconds, that an alarm can remain in the suppressed state. Enter 0 to disable the feature. The default is 1800. The T/EC adapter has a Suppress Alarm operation which suppresses any alarm with an OPEN status for a fixed amount of time. A suppressed alarm's severity level changes to SUPPRESSED and the alarm's severity does not populate up the element hierarchy. The alarm remains in this state until one of the following occurs: ◆ The operator closes the alarm ◆ The alarm changes severity to OK ◆ The timer expires ◆ The operator uses the Unsuppress Alarm operation
UseAlarmTimesFor CondChanges	The date/time stamp to use for all alarm data stored by the Operations Center Data Warehouse. If `true`, the alarm's date/time stamp is used. If `false`, the date/time stamp of when the Operations Center server received the alarm is used. The default is `true`.

- Section A.21.1, "Setup for Native "OCI" Oracle Driver," on page 325
- Section A.21.2, "Setup for DB2," on page 326

A.21.1 Setup for Native "OCI" Oracle Driver

Communicating with an Oracle version 7 or 8i (versions 8.1.7 and earlier) database requires installing the appropriate JDBC driver from Oracle. Operations Center does not natively connect to an Oracle 7 or 8i (versions 8.1.7 and earlier) database, nor to an Oracle database through the Oracle Client (OCI).

Oracle Releases Prior to 8.1.7: Formula 3.5 includes updated database drivers. These drivers do not support Oracle databases prior to version 8.1.7. Customers using an Oracle version prior to 8.1.7 with HP OpenView Operations for UNIX should upgrade to Oracle 8.1.7 or later. If this is not possible, contact Customer Support for other options.

To install the driver for Oracle 7 or 8i (8.1.7 and earlier):

1 Download the Oracle 7 or 8i driver and `orai18n.jar` file directly from Oracle.

Obtain a JDBC 1.1.1 driver for Oracle 7.

Oracle downloads are located at http://www.oracle.com/technology/software/tech/java/sqlj_jdbc/htdocs/winsoft.html.

2 Unzip the Oracle driver file to the */OperationsCenter_install_path*/classes directory.

3 Save the orai18n.jar file to the */OperationsCenter_install_path*/classes/ext directory.

A.21.2 Setup for DB2

Communicating with a DB2 database requires installing the appropriate JDBC driver from IBM. Operations Center does not natively connect to a DB2 database.

To enable TCP/IP communications, run an additional server program on the DB2 server. As noted in the DB2 documentation, to use the "net" version of the DB2 driver, a special server program, called db2jstrt, must run to allow a client to connect. Running this program with no arguments starts its listener on port 6789, which is the default setting for the DB2 driver. If this is not acceptable, change both ports.

To install the appropriate JDBC driver from IBM:

1 Locate the JDBC driver in the DB2 installation directory.

The file, db2java.jar, is usually found in the /sqllib/java directory.

2 Place the JAR file in */OperationsCenter_install_path*/classes/ext.

3 Restart Operations Center.

4 When configuring the adapter, enter db2 as the DB.type.

A.22 Mercury Application Mapping

Table A-26 *Mercury Application Mapping Adapter Properties*

Property	Specify…
Alarm Columns	A comma-separated list that determines which alarm columns display and the order in which the alarm items display (source of alarm, alarm class, etc.) in the *Alarms* view. A suggested list is: `assignedTo, description, status, lastModifiedBy, createDate, submitter`
Database	The name of the database that stores the Mercury Application Mapping repository.
Database Schema	The name of the Mercury Application Mapping schema in the database.
Elements Timeout	The number of seconds to display an element after all alarms have expired. Specify `-1` to never remove elements, `0` to remove elements immediately.
Hierarchy File	A file in the */OperationsCenter_install_path*/database directory that contains an XML description of the element hierarchy to build below the adapter element. The default is `examples/Mercury Application MappingHierarchy.xml`.
Hostname	The server where the Mercury Application Mapping database resides. Default is `localhost`.

Property	Specify…
Link Classes as Children A to B	When *Object Classes to Discover* is specified and in use, a comma-separated list of Mercury Application Mapping link classes to populate parent/child elements in Operations Center. Specify link classes that are to populate elements translating source endpoints as parent elements and destination endpoints as their children. These elements show in the adapter hierarchy under Elements. This setting is ignored if the *Root Object Classes to Discover* property is specified.
Link Classes as Children B to A	When *Object Classes to Discover* is specified and in use, a comma-separated list of Mercury Application Mapping link classes to populate parent/child elements in Operations Center. Specify link classes that are to populate elements translating destination endpoints as parent elements and source endpoints as their children. These elements show in the adapter hierarchy under Elements. This setting is ignored if the *Root Object Classes to Discover* property is specified.
Link Classes as Relationships A to B	When *Object Classes to Discover* is specified and in use, a comma-separated list of Mercury Application Mapping link classes to populate named relationships in Operations Center. Specify link classes that are to populate relationships with source endpoints remaining as the relationship source and destination endpoints remaining as the relationship destination. These relationships only show in the Relationship view and not in the adapter hierarchy. This setting is ignored if the *Root Object Classes to Discover* property is specified.
Link Classes as Relationships B to A	When *Object Classes to Discover* is specified and in use, a comma-separated list of Mercury Application Mapping link classes to populate named relationships in Operations Center. Specify link classes that are to populate relationships with source endpoints becoming the relationship destination and destination endpoints shown as the relationship source. These relationships only show in the Relationship view and not in the adapter hierarchy. This setting is ignored if the *Root Object Classes to Discover* property is specified.
Max Runtime Info Alarms	The maximum number of runtime information alarms to populate the adapter's Runtime Information branch under Elements. If set to any value `0` or below, no runtime information alarms are populated. Default is `-1`.
Object Classes to Discover	A comma-separated list of Mercury Application Mapping system object classes to be populated in the adapter's Network branch under Elements. This setting is ignored if the *Root Object Classes to Discover* property is set. Default is `host,ip,BusinessService,nt,unix,configfile,disk,file,interface,ipserver,memory,cpu`.
Password	The password for the user account to access the Mercury Application Mapping database.
Port	The port number on which the database host listens. Default is `1521`.
Root Objects Classes to Discover	A comma-separated list of Mercury Application Mapping root object classes to be populated in the adapter's Network branch under Elements. This setting takes precedence over and ignores setting in the *Object Classes to Discover* and *Link Classes* properties. Default is `BusinessService`.
Script.onError	A script that executes if the adapter fails for any reason. For example, the script can print the reason for the failure as msg using log.info(msg).
Script.onInitialized	A script that executes when the adapter initializes. All of the `Script.*` properties are optional.
Script.onStarted	A script that executes when the adapter starts, either manually or automatically when the Operations Center server starts.
Script.onStopped	A script that executes after manually stopping the adapter.

Property	Specify…
User Name	The user account to access the Mercury Application Mapping database. Default is `cmdb`.

A.23 Microsoft Operations Manager (MOM)

Table A-27 *Microsoft Operations Manager (MOM) Adapter Properties*

Property	Specify…
Alarm Columns	A comma-separated list that determines which alarm columns display and the order in which the alarm items display (date/time, rule, etc.) in the *Alarms* view. The default list is `Source`, `RepeatCount`, and `State`. These alarm columns are added to the base alarm properties such as Severity, Element, and so on.
Configuration File	This file is for SQL queries and column strings. The default file name is `/`*`OperationsCenter_install_path`*`/database/msmomConfiguration.xml`. If a different file name is specified but does not exist, it is created upon adapter startup. If the configuration DTD file does not exist in the directory, then it is written there when the adapter is started. You might need to adjust the Severity Remappings (Numeric) adapter property appropriately for the MOM 2005 or MOM 2007 severity mappings.
Configuration Version	The database version; can be either MOM2005 or MOM2007 and determines the properties to use from the configuration file.
Database Hostname	The name of the SQL server on which the MOM database is installed.
Database Login ID	The login ID for the adapter to connect to the database. The default is `sa`.
Database Name	The name of the database that stores the MOM repository. The default is `Onepoint` for MOM and `OperationsManager` for SCOM.
Database Password	The password that corresponds to ID specified in the Database Login property.
Database Port	The port on which the SQL Server listens for database connections. The default is 1433.
Domain	SQL Server provides authentication based on Windows accounts and a named SQL server login ID and password. If SQL Server is configured for Windows Only authentication mode, then you must provide the domain name as well as the login ID and password in the adapter properties. If SQL Server is configured for SQL Server and Windows authentication mode, then providing the domain name is optional. Either provide the user name, password, and domain in the adapter properties, or provide the SQL login ID and password and leave the domain empty.

Property	Specify…
Elements Timeout	If there are no open alarms and the element's condition hasn't changed in the last n seconds, and if the element has no children, then the element disappears from the display. If another alarm is generated for this element, then it reappears. The default is `-1`, elements are never removed even if they have no condition changes or alarms. Set the Elements Timeout property value using one of the following logic: **ElementsTimeout < 0:** Never time out. **ElementsTimeout = 0:** Time out immediately. **ElementsTimeout > 0:** Time out after specified time expires. If set to 0 or a number greater than 0, the initial population of computers is not performed at adapter start up; therefore, the adapter does not show any elements without alarms.
Event Viewer Exec Command	Command used to start the MS Windows Event Viewer. If no value is specified, the following command is used: `cmd.exe /C eventvwr.exe` The MS Windows Event Viewer can only be run on client machines running MS Windows. The Event View might display "access or permission denied" message if the Windows user does not have sufficient permissions to connect to the remote computer. If this occurs, contact your Windows Administrator.
Hierarchy File	A file in the */OperationsCenter_install_path/*`database` directory that contains an XML description of the element hierarchy to build below the adapter element. The default is `examples/msmom.xml`.
Polling Interval (Seconds)	The number of seconds between queries for new alarms or re-queries for updating existing alarms. The default is 5.
Script.onError	A script that executes if the adapter fails for any reason. For example, the script can print the reason for the failure as msg using log.info(msg).
Script.onInitialized	A script that executes when the adapter initializes.
Script.onStarted	A script that executes when the adapter starts, either manually or automatically when the Operations Center server starts.
Script.onStopped	A script that executes after manually stopping the adapter.
Seed File	The seed file by default is blank, so there are no seed alarms. If a file name is provided and it does not exist in the directory given (the base directory is the */OperationsCenter_install_path/*`database` directory), then an example file is created when the adapter is started. If the seed file DTD file does not exist in the directory, then it is written there when the adapter is started.

Property	Specify…
Severity Remappings	It is necessary to translate the severity levels in MOM to the severity codes used in Operations Center. Operations Center allows the following severities: ◆ OK (usually green) ◆ INFORMATIONAL (usually blue) ◆ MINOR (usually yellow) ◆ MAJOR (usually orange) ◆ CRITICAL (usually red) ◆ UNKNOWN (usually gray) The MOM severity definitions are discovered at startup from the Onepoint.ResolutionState table. This mapping process uses the contents of the State column. The process requires mapping a MOM state (such as Success) to a Operations Center severity (such as OK). For example: `Success=OK` The default mappings are: `Success=OK:Information=INFORMATIONAL:Warning=MINOR:Error=MAJOR:Critical Error=CRITICAL:Security Breach=CRITICAL:Service Unavailable=CRITICAL` For MOM 2005: `10=OK:20=INFORMATIONAL:30=MINOR:40=MAJOR:50=CRITICAL:60=CRITICAL:70=CRITICAL` For MOM 2007, change to: `1=OK:0=INFORMATIONAL:2=MINOR:2=MAJOR:2=CRITICAL:3=CRITICAL:4=CRITICAL`
Stylesheet File	The stylesheet file in the */OperationsCenter_install_path/*`database` directory that applies to the HierarchyFile as a style markup to produce the final output.
Timeout	Specifies a socket timeout, in seconds, for database communications. If there is a faulty network connection, the database communications times out after this period. Default is `0`. If database queries are running longer before results are returned, it is best to set a high enough socket timeout to alleviate the risk of the timeout being triggered while results are still being gathered on the database server. However, we recommend that you don't set the socket timeout value too high because if the connection is unexpectedly dropped, the next query might not be executed until the connection is re-established and the timeout value is reached.
Use Alarm Times For Condition Changes	The date/time stamp to use for all alarm data stored by the Operations Center Data Warehouse. If `true`, the alarm's date/time stamp is used. If `false`, the date/time stamp of when the Operations Center server received the alarm is used. The default is `true`. Alarm history is stored based on the alarm time rather than alarm receipt time. Also, for SLA metric data based on alarm properties, the property values are recorded based on the alarm time instead of the alarm receipt time. Note that recording historical condition data for historical alarms is not supported.

A.24 Microsoft System Center Operations Manager (SCOM)

Table A-28 *Microsoft System Center Operations Manager (SCOM) Adapter Properties*

Property	Specify…
Alarm Columns	A comma-separated list that determines which alarm columns display and the order in which the alarm items display (date/time, rule, etc.) in the *Alarms* view. The default list is `Source`, `RepeatCount`, and `State`. These alarm columns are added to the base alarm properties such as Severity, Element, and so on.
Elements Timeout (Seconds)	If there are no open alarms and the element's condition hasn't changed in the last n seconds, and if the element has no children, then the element disappears from the display. If another alarm is generated for this element, then it reappears. The default is `300`. Set the Elements Timeout property value using one of the following logic: **ElementsTimeout = -1:** Never time out. **ElementsTimeout = 0:** Time out immediately. **ElementsTimeout > 0:** Time out after specified time expires. If set to 0 or a number greater than 0, the initial population of computers is not performed at adapter start up; therefore, the adapter does not show any elements without alarms. If *Integration Type* is set to `Event`, consider setting *Element Timeout* to `0` to show only the computers with events.
Hierarchy File	A file in the */OperationsCenter_install_path/*`database` directory that contains an XML description of the element hierarchy to build below the adapter element. The default is `examples/DefaultSCOMEventBasedHierarchy.xml`. This property must be left blank when *Integration Type* is set to `Object`.
Integration Type	The type of integration. Set to: `Object`, `Event`, or `Both`. Default is `Both`. ◆ **Event:** Creates the object hierarchy based on the hierarchy file defined in the *Hiearchy File* field. ◆ **Object:** Creates the object hierarchy based on the hierarchy found in SCOM. ◆ **Both:** Creates the object hierarchy based on the hierarchy file and the SCOM hierarchy. When setting to `Object`, be sure no value is specified (left blank) for the Hierarchy File property.
Polling Interval (Seconds)	The number of seconds between queries for new alarms or re-queries for updating existing alarms. The default is `60`. Set to `0` (to disable polling), or a number greater than `0`. If property is left blank, the default is used.
SCOM Domain	The domain name for the SCOM server.
SCOM Server (DNS name or IP address	The IP address or fully-qualified domain name for the SCOM server.
SCOM Username	The user name for the SCOM administrator account.
SCOM Username Password	The password for the SCOM user account.

Property	Specify…
Script.onError	A script that executes if the adapter fails for any reason. For example, the script can print the reason for the failure as msg using log.info(msg).
Script.onInitialized	A script that executes when the adapter initializes.
Script.onStarted	A script that executes when the adapter starts, either manually or automatically when the Operations Center server starts.
Script.onStopped	A script that executes after manually stopping the adapter.
Severity Remappings	It is necessary to translate the severity levels in SCOM to the severity codes used in Operations Center. Operations Center allows the following severities: ◆ OK (usually green) ◆ INFORMATIONAL (usually blue) ◆ MINOR (usually yellow) ◆ MAJOR (usually orange) ◆ CRITICAL (usually red) ◆ UNKNOWN (usually gray) The default mappings are: `Success=OK:Information=INFORMATIONAL:Warning=MINOR:Error=CRITICAL:Uninitialized=UNKNOWN`
Stylesheet File	The stylesheet file in the */OperationsCenter_install_path/*`database` directory that applies to the HierarchyFile as a style markup to produce the final output.
Use Alarm Times For Condition Changes	The date/time stamp to use for all alarm data stored by the Operations Center Data Warehouse. If `true`, the alarm's date/time stamp is used. If `false`, the date/time stamp of when the Operations Center server received the alarm is used. The default is `true`. Alarm history is stored based on the alarm time rather than alarm receipt time. Also, for SLA metric data based on alarm properties, the property values are recorded based on the alarm time instead of the alarm receipt time. Note that recording historical condition data for historical alarms is not supported.

A.25 NetIQ AppManager

Table A-29 *NetIQ AppManager Adapter Properties*

Property	Specify…
AlarmColumns	A comma-separated list that determines which alarm columns display and the order in which the alarm items display (date/time, rule, etc.) in the *Alarms* view. Available alarm columns include: *EventID*, *ParentEventID*, *JobID*, *Status*, *FirstOccurConsoleTime*, *LastOccurConsoleTime*, *FirstOccurTimeString*, *LastOccurTimeString*, *ObjID*, *MachineName*, *KPName*, *Severity*, *EventMsg*, *Occurrence*, *ChildComment*, and *ModificationTime*.

Property	Specify…
ControlCenterExtensions	Enables special extensions used with AppManager Control Center and enables the SCM job to communicate with the adapter. Default is `false`. Set this property to `true` only if you are running the NetIQ AppManager Operations Portal integration. For more information, see the *NetIQ AppManager Operations 5.5 Portal Getting Started Guide*.
DbHost	The name of the host on which the NetIQ data repository resides.
DbName	The NetIQ database name. The default is `QDB`. If the NetIQ database is located on a named instance of a database, declare the instance name after the database name, and separated by a semi-colon: `QDB;instance=`*`instance_name`* For example, `QDB;instance=SQL208R2`
DbPassword	The DbUser password.
DbPort	The port on which the NetIQ database server listens. The default is 1433 which is the standard SQL Server port configuration.
DbProperties	Enter a value when it is necessary to override a database connection property. For example, some versions of SQL Server require setting AutoCommit to False, but the default setting in DbProperties is `AutoCommit=true`. Use DbProperties to set it to `false`. To specify multiple properties, comma-delimit the name-value pairs.
DbUser	The database user ID with unrestricted access to the database identified in the DbName property. The default ID is `sa`.
DiscoveryViews	A list of views to discover at adapter startup. The default is `Master`, which allows the discovery of all views listed in the master. Add any additional custom views separated by a comma.
ReconTimer	The NetIQ adapter schedules reconciliation events every x number of minutes following the completion of the previous reconciliation event. Use the Reconciliation Time property to specify the number of minutes for the time interval. For example, if it takes 30 minutes to reconcile the differences each time and the schedule for the ReconTimer is 15 minutes, the entire reconciliation event actually takes 45 minutes. The default is 0.
ReformatEventFields	Alarm column names that require Operations Center to truncate leading characters. For example, if the KPName column contains ###:NT_CPU but only NT_CPU should display, the entry is: `KPName=:` Use any characters as the truncation delimiter. If more then one alarm column requires truncation, use a comma to separate the field name and delimiter value pairs. For example: `KPName=:,ChildComment=&.`
Script.onError	A script that executes if the adapter fails for any reason. For example, the script can print the reason for the failure as msg using log.info(msg).

Property	Specify…
Script.onInitialized	A script that executes when the adapter initializes.
Script.onStarted	A script that executes when the adapter starts, either manually or automatically when the Operations Center server starts.
Script.onStopped	A script that executes after stopping the adapter.
ScriptCategories	The list of Knowledge Scripts (KS) that run to populate element jobs. The default categories are: `ACTION,ARCSERVE,CLIENT,DISCOVERY,GENERAL,MTS,NT,NTADMIN,SQL,WIN2000.`

A.26 NetIQ Cloud Manager

Table A-30 *NetIQ Cloud Manager Adapter Properties*

Property	Specify…
Alarm Columns	A comma-separated list that determines which alarm columns display and the order in which the alarm items display (source of alarm, alarm class, etc.) in the *Alarms* view. This property is not used by the NetIQ Cloud Manager adapter as there are no alarms.
NetIQ Cloud Manager Server	IP address of the NetIQ Cloud Manager Server.
NetIQ Cloud Manager User Name	The NetIQ Cloud Manager Administrator user name.
NetIQ Cloud Manager User Password	The NetIQ Cloud Manager Administrator password.
Poll Interval (mins)	The interval, in minutes, that the adapter performs an automatic full refresh of the hierarchy. Defaults to 5.
Port	The port on which the NetIQ Cloud Manager Server listens. Defaults to 8182.
Script.onError	A script that executes if the adapter fails for any reason. For example, the script can print the reason for the failure as `msg` using `log.info(msg)`.
Script.onInitialized	A script that executes upon adapter initialization. All of the `Script.*` properties are optional.
Script.onStarted	A script that executes when the adapter starts, either manually or automatically when the Operations Center server starts.
Script.onStopped	A script that executes after manually stopping the adapter.
Use NCSS	Set to `true` to use the `?admin` option to authenticate via NCSS and LDAP. Defaults to `false`.
Use SSL	Set to `true` to use SSL to connect to the NetIQ Cloud Manager server. Defaults to `false`.

A.27 NetIQ Sentinel

Table A-31 *NetIQ Sentinel*

Property	Specify
Alarm Columns	A comma-separated list that determines which alarm columns display and the order in which the alarm items display in the Alarms view. ***Sentinel 6 Adapter:*** Default is `Event Name, Event Created, Rule Name, Rule Created, Route, Parent Id` ***Sentinel 6 Adapter:*** Available columns are `Event Name, Event Id, Message, Parent Id, Rule Name, Alarm Created Long, Event Created Long, Rule Created Long, Event Create, Rule Created, Occurrences, Host, Source IP, Severity, Route`. ***Sentinel 7Adapter***: Default is `Event Name, Message, Rule Name, Route, Parent Id` ***Sentinel 7 Adapter:*** Available columns are `Event Name, Event Id, Message, Parent Id, Rule Name, Alarm Created Long, Event Created Long, Occurrences, Host, Source IP, Severity, Route, Reporter IP, TenantName, ObserverCountry, SourceHostCountry, TargetHostCountry, TargetHostLatitude, TargetHostLongitude, SourceHostLatitude, SourceHostLongitude, ObserverHostLatitude, ObserverHostLongitude, ObserverHostName, ObserverIP, ObserverHostCountry, ObserverServiceName, SourceHostName, TargetHostName, TargetIP, TargetUserFullName, TargetEmail, InitiatorEmail, InitiatorUserFullName, Vulnerability, XDASOutcomeName, XDASTaxonomyName, CollectorNodeName, InitiatorUserName, TargetUserName, SourceHostDomain, InitiatorUserDomain, TargetHostDomain, TargetUserDomain, InitiatorServiceComponent, TargetServiceComponent, TargetServiceName, InitiatorServiceName, TargetTrustDomain, TargetTrustName, TargetDataContainer, TargetDataName`. For information about Sentinel
Alarm Expiration Polling Time (in Seconds)	The interval, in seconds, that the time stampadapter performs an evaluation of alarms and remove expired alarms.
Alarm Expiration Time Type	The time stamp to use when evaluation alarms. Specify one of the following: ◆ **event:** the time of the event in Sentinel ◆ **rule:** ***Sentinel 6 Adapter Only.*** the time the Correlation Rule was triggered in Sentinel ◆ **alarm:** the time the alarm was received in Operations Center
Critical Max	The highest value that can be mapped to a Critical event. Default is 5.

Property	Specify
Custom Property Mappings	***Sentinel 7 Adapter:*** A comma delineated list of name/value pairs for declaring custom Sentinel event properties. When configured, these properties show in a *Custom Attributes* alarm property page. Use the following syntax to map event fields to an alarm property: `Custom_`*`alarm_property_name`*`=`*`sentinel7_event_field_name`* For example, `Custom_Customer Source IP=dip, Custom_Customer Source=rv39`. The `Custom_` prefix is used in Operations Center to avoid property name clashes and can be omitted from the property definition. However, the actual alarm property retains the full property name. For example, `Custom_Customer Source` If adding to the *Alarm Columns* list in adapter properties, the full property name must be specified, but can be mapped to a shorter name for display purposes. For example, `Customer Source=Custom_Customer Source` If using in the hierarchy file to generate new elements in the Sentinel adapter hierarchy tree from property values, the full property name must be specified. For example, `<generator class="SentinelHost" field="Custom_Customer Source"/>`
Element Timeout Ager Delay Value (in Seconds)	The number of seconds to display an element after all alarms have expired. Specify `-1` to never remove elements, `0` to remove elements immediately.
Hierarchy File	A file in the /*`OperationsCenter_install_path`*`/database` directory that contains an XML description of the element hierarchy to build below the adapter element. The default is `examples/SentinelHierarchy.xml`
Incoming Event Thread Pool Size	The number of threads that can be started for processing incoming events from Sentinel. This can be useful for performance tuning / resource management.
Info Max	The highest value that can be mapped to an Information event.
Listener Port	The Port number to be opened for incoming events. ***Sentinel 6 Adapter:*** Set to the same port number as the Sentinel Mail/SMTP Interceptor port. ***Sentinel 7 Adapter***: Set to the same port number as the Sentinel *Log to Syslog* Action port.
Major Max	The highest value that can be mapped to a Major event.
Minor Max	The highest value that can be mapped to a Minor event.
Rule List for History Mining	The names of Correlation Rule recent events to be queried on adapter startup. ***Sentinel 6 Adapter***: Specify a comma delimited list using single quotes around the values. For example, `'Test Rule 1', 'Test Rule 2', 'Test Rule 3'` ***Sentinel 7 Adapter:*** Specify a comma delimited list. For example, `Test Rule 1, Test Rule 2, Test Rule 3`
Script.onError	A script that executes if the adapter fails for any reason. For example, the script can print the reason for the failure as `msg` using `log.info(msg)`.
Script.onInitialized	A script that executes when the adapter initializes.

Property	Specify
Script.onStarted	A script that executes when the adapter starts, either manually or automatically when the Operations Center server starts.
Script.onStopped	A script that executes after stopping the adapter.
Sentinel Database Connection Password	***Sentinel 6 Adapter Only***. The password used to connect to the Sentinel database for history mining.
Sentinel Database Connection User Name	***Sentinel 6 Adapter Only.*** The Account Name used to connect to the Sentinel database for history mining.
Sentinel Database Name	***Sentinel 6 Adapter Only***. The name of the Sentinel database for history mining (Oracle database name).
Sentinel Database Server Address	***Sentinel 6 Adapter Only***. The IP address of the Sentinel database for history mining (Oracle database name).
Sentinel Database Server Port	***Sentinel 6 Adapter Only***. The port for connecting to the Sentinel Database for history mining.
Sentinel Database Time Zone Offset	***Sentinel 6 Adapter Only***. The number of hours off of GMT to be used for the Sentinel database connection. For example, enter 7 for MST.
Sentinel Server Administrator User Password	***Sentinel 7 Adapter Only***. The Administrator User password used to connect to Sentinel for history mining.
Sentinel Server Administrator User Name	***Sentinel 7 Adapter Only.*** The Administrator User account name used to connect to Sentinel for history mining.
Sentinel Server Address	***Sentinel 7 Adapter Only.*** The IP address of the Sentinel server for history mining.
Sentinel Server Port	***Sentinel 7 Adapter Only.*** The port for connecting to the Sentinel REST interface for history mining.
Stylesheet File	This option is not used by the Sentinel adapter.
Time Length in Minutes/ Hours for Events to Display	The number of minutes/hours an alarm is displayed before it is removed from the console. ***Sentinel 6 Adapter***: Specify number of minutes. ***Sentinel 7 Adapter***: Specify number of hours.
Time Length in Minutes/ Hours for History Mining	The number of minutes/hours of history from the Sentinel database (***Sentinel 6 Adapter***) or Sentinel server (***Sentinel 7 Adapter***) to be loaded on adapter startup. ***Sentinel 6 Adapter***: Specify number of minutes. ***Sentinel 7 Adapter***: Specify number of hours.
Use Alarm Times For Condition Changes	The date/time stamp to use for all alarm data stored by the Operations Center Data Warehouse. If `true`, the alarm's date/time stamp is used. If `false`, the date/time stamp of when the Operations Center server received the alarm is used. The default is `true`. Alarm history is stored based on the alarm time rather than alarm receipt time. For SLA metric data based on alarm properties, property values are recorded based on the alarm time instead of the alarm receipt time. Note that recording historical condition data for historical alarms is not supported.

A.28 NetIQ Operations Center Experience Manager

For information about the Experience Manager adapter properties, see "Creating a Experience Manager Adapter" in the *Operations Center 5.5 Experience Manager Guide*.

A.29 NetIQ Operations Center Event Manager

For information about the Experience Manager adapter properties, see "Creating an Event Manager Adapter" in the *Operations Center 5.5 Event Manager Guide*.

A.30 NetIQ Operations Center F/X

For information about the Experience Manager adapter properties, see "Defining the Adapter Properties" in the *Operations Center 5.5 F/X Adapter Guide*.

A.31 NetIQ Operations Center InterConnection

Table A-32 *Operations Center InterConnection Adapter Properties*

Property	Specify…
CorrectTimeSkew	An option used to synchronize date/time stamps between server clocks. If True, all date/time stamps are adjusted to reflect the time differences between two Operations Center servers. If False, all date/time stamps are displayed as received from the remote server and no adjustments are made for time differences between two servers. The default is True.
DiscoverAdministration	If True, the Administration branch on a remote server is accessible on the connected machine. If False, the Administration branch is not accessible. For details on the administration tasks that can be performed remotely, see the *Operations Center 5.5 Server Configuration Guide*.
DiscoverAll	If False, discovery of remote elements occurs as they are revealed. If True, discovery of remote elements occurs all at once at adapter startup. The default is True.
LogUnresolvedRemotes	A log control mechanism. An attempt is made to discover all InterConnection adapter (ICA) elements linked from Business Service Views. If True, logs warnings if it cannot discover ICA elements. If False, disables logging. The default is False.
LoginAccount	The remote Operations Center server user name. The default is `admin`.
LoginPassword	The user name's password. The default is `formula`.
Script.onError	A script that executes if the adapter fails for any reason. For example, the script can print the reason for the failure as msg using log.info(msg).
Script.onInitialized	A script that executes when the adapter initializes. All of the `Script.*` properties are optional.
Script.onStarted	A script that executes when the adapter starts, either manually or automatically when the Operations Center server starts.
Script.onStopped	A script that executes after manually stopping the adapter.

Property	Specify…
ServerURL	The URL used to reach the Operations Center server, with /Reference appended. For unsecured TCP/IP connection on UNIX, the default port is 8080. Assume the Operations Center server is on a host named global. The value for ServerURL is http://global:8080/Reference. To establish a secure connection on UNIX using SSL, assuming the server accepts HTTPS connections on the default port 8443, the value for ServerURL is https://global:8443/Reference. For Windows, use 80/443 instead of 8080/8443. For more information on using unsecure and secure client/server communication, see the *Operations Center 5.5 Security Management Guide* and the *Operations Center 5.5 Server Configuration Guide*.
ShowRemote Operations	If True, uses all operations from the remote element. If False, uses only the element and alarm operations from the local Operations Center server. The default is True.

A.32 NetIQ Operations Center Universal

Table A-33 *NetIQ Operations Center Universal Adapter Properties*

Property	Specify…
AlarmColumns	A comma-separated list that determines which alarm columns display and the order in which the alarm items display in the *Alarms* view. The defaults are: `Status`, `Class`, and `Description`. Aliases for Alarm Columns: Assign new names to alarm columns using the format: *display_name=current_name*. For example, `Condition=Status` displays status data in a column named *Condition*.
ClosedAlarmsTimeout	The length of time, in seconds, to display an alarm after closing it in the Operations Center console. Enter -1 to display the alarm indefinitely. Enter 0 to remove it immediately. The default is 1800.
DuplicateCount	The event slot name containing the number of times that an alarm was received. Use when it is possible to duplicate alarms based on the originating_event_id.
ElementsTimeout	If there are no open alarms, and the element's condition hasn't changed in the last n seconds, and the element has no children, then the element disappears from the display. If another alarm is generated for this element, then it reappears. The default is 300. **ElementsTimeout < 0:** Never time out. **ElementsTimeout = 0:** Time out immediately. **ElementsTimeout > 0:** Time out after specified time expires.
EventConsoleName	The ID used by the Operations Center server to identify itself to the Script adapter. This value must be a valid client name of the system being integrated. The default is `@Formula`.

Property	Specify…
EventListenPort	The TCP/IP socket port number to which the adapter forwards its events. Use any port number (above 1000 in UNIX). The default is 54321.
HierarchyFile	A file in the */OperationsCenter_install_path/*database directory that contains an XML description of the element hierarchy to build below the adapter element. The default is examples/ScriptHierarchy.xml.
MaxAlarms	The maximum number of alarms that the adapter queries and retains. Exceeding this number removes the oldest alarm. Enter 0 to allow an unlimited number of alarms. The default is 500.
Post status changes to TEC	Attempts to send a postem message back to host when a status change occurs on an alarm having originating_tec_id and originating_tec_hostname values defined. If set to false, no messages are sent. Defaults to true. For additional information, see Section 7.3.2, "Data Fields," on page 214 and Section 7.6, "Understanding Alarm Operations and Event Status," on page 217.
Script.onError	A script that executes if the adapter fails for any reason. For example, the script can print the reason for the failure as msg using log.info(msg).
Script.onInitialized	A script that executes when the adapter initializes. All of the Script.* properties are optional.
Script.onStarted	A script that executes when the adapter starts, either manually or automatically when the Operations Center server starts.
Script.onStopped	A script that executes after stopping the adapter.
SeedFile	A file that contains information about managed elements, which enables the Operations Center server to generate the elements hierarchy.
StylesheetFile	The stylesheet file in the */OperationsCenter_install_path/*database directory that applies to the HierarchyFile as a style markup to produce the final output.
SyncClass	The default is Sync, which allows multiple instances of the adapter to synchronize their alarms.
UpdateTimestamp	The name of the incoming event slot containing the alarm date. If no value is specified, the alarm date/time is the date/time that the alarm was received.
UseAlarmTimesForCondChanges	The date/time stamp to use for all alarm data stored by the Operations Center Data Warehouse. If true, the alarm's date/time stamp is used. If false, the date/time stamp of when the Operations Center server received the alarm is used. The default is true.

A.33 NetIQ Operations Center SNMP Integrator

For information about the Experience Manager adapter properties, see "Creating an SNMP Adapter" in the *Operations Center 5.5 SNMP Integrator Guide*.

A.34 NetIQ Sentinel

For information about the Novell Sentinel adapter properties, see Section A.27, "NetIQ Sentinel," on page 335.

A.35 Novell ZENworks

Table A-34 *Novell ZENworks*

Property	Specify
Alarm Columns	A comma-separated list that determines which alarm columns display and the order in which the alarm items display in the Alarms view. Default is `occurrences, description`.
Database Timeout (minutes)	Specify how long the adapter waits (in minutes) for a SQL query to return before timing out. Set this value to a higher value if your zone is so large that a single SQL query does not return before the timeout value is reached. Higher values result in a longer timeout if the database cannot be contacted. Default is 15.
Script.onError	A script that executes if the adapter fails for any reason. For example, the script can print the reason for the failure as `msg` using `log.info(msg)`.
Script.onInitialized	A script that executes when the adapter initializes.
Script.onStarted	A script that executes when the adapter starts, either manually or automatically when the Operations Center server starts.
Script.onStopped	A script that executes after stopping the adapter.

A.36 PlateSpin Orchestrate

Table A-35 *PlateSpin Orchestrate*

Property	Specify
Alarm Columns	A comma-separated list that determines which alarm columns display and the order in which the alarm items display in the Alarms view.
PSO Custom VM Facts	Specify one or more fact names and label pairs in a comma deliminated list to display on the VM object's property page using the following syntax: *`resource.factname:label, resource2.factname2:label2`* Where `factname` is the full factname for a VM, and the value is displayed on the object's properties page. The propery name is formed by using CFACT-*label*. For example, if we specified `resource.disktype:Disk Type` then the property name displayed would be `CFACT-Disk Type` and the value would be the value of `resource.disktype`, such as `scsi`. To store the fact for correlation purposes without displaying on the properties page, add square brackets around the *`label`*. For example: *resource.factname:[label]*

Property	Specify
PSO Disk Percent Used Threshold (critical)	The percentage of repository disk space used that generates a critical alarm. For example, 95.
PSO Disk Percent Used Threshold (major)	The percentage of repository disk space used that generates a major alarm. For example, 90.
PSO Disk Percent Used Threshold (minor)	The percentage of repository disk space used that generates a minor alarm. For example, 80.
PSO Poll Interval (mins)	The interval at which the adapter performs an automatic full refresh of the hierarchy.
PSO User Name	The PSO admin account. A default install of PSO creates the `zosadmin` account. If a new admin account has been created, use the new account.
PSO Password	The password for the PSO admin account.
PSO Web Service URL	The Web service URL using the fully-qualified domain name for the PSO Server, such as: `https://PSO_Server_DNS_Name:port_number/PSORest/` For example, `https://mycompanydnsname.com:8443/PsoRest/`
Script.onError	A script that executes if the adapter fails for any reason. For example, the script can print the reason for the failure as `msg` using `log.info(msg)`.
Script.onInitialized	A script that executes when the adapter initializes.
Script.onStarted	A script that executes when the adapter starts, either manually or automatically when the Operations Center server starts.
Script.onStopped	A script that executes after stopping the adapter.
Show Stopped VMs as a Critical Condition	If `true`, stopped VMs show a condition of Critical. Defaults to `false`.

A.37 PlateSpin Recon

Table A-36 *PlateSpin Recon Adapter Properties*

Property	Specify…
Alarm Columns	A comma-separated list that determines which alarm columns display and the order in which the alarm items display in the *Alarms* view. Available columns are *ID*, *Element*, *Severity*, *Date/Time*.
Database	The name of the PlateSpin Application database. Default database name is `platespin_powerrecon_36_application`.
Database Type	The type of repository. Only valid value is `mssql`. Only MS SQL 2000 and 2005 are supported databases for the PlateSprin Recon adapter.
Hostname	The name of the server on which the PlateSpin Recon repository resides.
Monitoring Database	The name of the PlateSpin Monitoring database. Default database name is `platespin_powerrecon_36_monitoring`.
Password	The user name to log in to the PlateSpin Recon repository.
Port	The port for the PlateSpin Recon repository. The default port for MSSQL is 1433.
Query Schedule – Applications	Sets the schedule to query for data regarding any applications. Disabled by default. Specify a query schedule using Macro Expressions for Query Schedules.
Query Schedule – Logs	Sets the schedule to query for log data. Disabled by default. Specify a query schedule using Macro Expressions for Query Schedules. **WARNING:** Because a large amount of data is often retrieved when scheduling for logs is active, memory errors can occur in the console *Layout* view. If this happens, increase memory settings in the Configuration Manager. For more information on the Configuration Manager, see the *Operations Center 5.5 Server Configuration Guide*.
Query Schedule – Processes	Sets the schedule to query for data regarding any processes. Disabled by default. Specify a query schedule using Macro Expressions for Query Schedules.
Query Schedule – Services	Sets the schedule to query for data regarding any services. Disabled by default. Specify a query schedule using Macro Expressions for Query Schedules.
User Name	The password to log in to the PlateSpin Recon repository.

- Section A.37.1, “Macro Expressions for Query Schedules,” on page 343

A.37.1 Macro Expressions for Query Schedules

Because the PlateSpin Recon Adapter is built on the Data Integrator platform, query schedule definitions use the same macro expressions as Data Integrator.

For additional information, see “Scheduling Queries” in the *Operations Center 5.5 Data Integrator Guide*.

Macro expression schedules can use either of the following formats:

- [setting1]; [setting2]; [setting3]; ...
- [setting1] & [setting2] & [setting3] & ...

Settings are not case-sensitive. Table A-37 lists acceptable values for query schedules:

Table A-37 *Acceptable Values for Query Schedules*

Setting	Description
`disable (aliases are 'disabled', 'off', 'none', 'false')`	If present, the schedule does not fire regardless of the other settings.
`utc`	If present, then dates are interpreted as being in UTC format.
`atstart (aliases are 'at start', 'onstart', 'on start')`	If present, the schedule fires upon adapter startup.
`atevent (aliases are 'at event', 'onevent', 'on event')`	If present, then the schedule can be triggered by a scriptable event as described previously.
`every n seconds|minutes|hours [between HH:MM and HH:MM] [on MON,TUE,WED,THU,FRI,SAT,SUN]`	Where n is a number representing the interval, which can be seconds, minutes or hours (abbreviate using the first letter). The between times and the on days clauses are optional. HH hours are always in 24-hour notation as AM and PM are not allowed. Examples: ◆ every 5m ◆ every 1h between 20:00 and 23:00 ◆ every 1h on Mon,Wed,Fri ◆ every 1 hour ◆ every 5 hours between 10:00 and 16:00 on Sat,Sun
`at HH:MM, HH:MM, HH:MM [on MON,TUE,WED,THU,FRI,SAT,SUN]`	Where the on days clause is optional. HH hours are always in 24-hour notation as AM and PM are not allowed. Examples: ◆ at 10:00 ◆ at 10:45, 22:45 ◆ at 13:30, 16:30 on Mon,Wed,Fri

A.38 SolarWinds Orion Adapter

Table A-38 *SolarWinds Orion Adapter Properties*

Property	**Specify…**
Alarm Columns	A comma-separated list that determines which alarm columns display and the order in which the alarm items display (source of alarm, alarm class, etc.) in the *Alarms* view. A suggested list is: `assignedTo, description, status, lastModifiedBy, createDate, submitter`
Database	The name of the database on the SQL server. The default is `NetPerfMon`.

Property	Specify…
Hostname	The hostname of the SQL server.
Max Alarms	The maximum number of alarms that the adapter queries and retains. Enter 0 to allow an unlimited number of alarms. The default is 10000.
Password	The password for the database account.
Port	The port number on which the SQL server listens. The default is 1433.
Script.onError	A script that executes if the adapter fails for any reason. For example, the script can print the reason for the failure as msg using log.info(msg).
Script.onInitialized	A script that executes when the adapter initializes.
Script.onStarted	A script that executes when the adapter starts, either manually or automatically when the Operations Center server starts.
Script.onStopped	A script that executes after manually stopping the adapter.
Show Applications	Remove the `disabled` value (set to blank) to show applications from SolarWinds in the adapter. Defaults to `disabled`.
User Name	The user name for the database account on the SQL server.
The following optional properties relate to polled updates for “map” elements in SolarWinds, which are viewable (but not editable) in the Operations Center console *Layout* view. The server polled is the Microsoft IIS server that provides Web dashboard views on the SolarWinds server. The IIS server must allow remote access, and the account you supply must be able to view maps in SolarWinds Orion.	
WebAccountName	The account name for the Web dashboard account.
WebAccountPassword	The password for the Web dashboard account.
WebQueryInterval	The number of seconds to poll for updates on the Web dashboard for map data (map names, nodes on map). The default is 300.
WebServerHost	The hostname for Web dashboard.
WebServerPort	The host port for Web dashboard. The default is 80.

A.39 Symantec Clarity

Table A-39 *Symantec Clarity Adapter Properties*

Property	Specify…
Alarm Columns	A comma-separated list that determines which alarm columns display and the order in which the alarm items display (source of alarm, alarm class, etc.) in the *Alarms* view. A suggested list is: `assignedTo, description, status, lastModifiedBy, createDate, submitter`

Property	Specify…
Elements Timeout	The length of time, in seconds, to age out elements. If no open alarms exist and the element's condition does not change in n seconds, and the element has no children, then the element disappears. The element redisplays if another alarm is generated. The default is 300 seconds. **AgeOutTime < 0:** Never age out. **AgeOutTime = 0:** Age out immediately. **AgeOutTime > 0:** Age out after specified time expires.
Hierarchy File	A file in the */OperationsCenter_install_path/*`database` directory that contains an XML description of the element hierarchy to build below the adapter element. The default is `examples/Symantec ClarityHierarchy.xml`.
Hostname	The hostname for the Symantec Clarity server.
Max Alarms	The maximum number of alarms that the adapter queries and retains. Enter 0 to allow an unlimited number of alarms. The default is 1000.
Max Alarms Per Query	The maximum number of alarms to retrieve per query. The default is 1000.
Password	The password for database account.
Port	The port number on which the database listens. The default is 1521.
Script.onError	A script that executes if the adapter fails for any reason. For example, the script can print the reason for the failure as msg using log.info(msg).
Script.onInitialized	A script that executes when the adapter initializes.
Script.onStarted	A script that executes when the adapter starts, either manually or automatically when the Operations Center server starts.
Script.onStopped	A script that executes after manually stopping the adapter.
Server Id	The server ID or database name.
User Name	The user name for the database account.

A.40 Tideway Foundation

Table A-40 *Tideway Foundation Adapter Properties*

Property	Specify…
Alarm Columns	A comma-separated list that determines which alarm columns display and the order in which the alarm items display (source of alarm, alarm class, etc.) in the *Alarms* view. A suggested list is: `assignedTo, description, status, lastModifiedBy, createDate, submitter`
Auto Create Relationships	Controls the automatic creation of the relationships among the four component types (BAI, HOST, SE, SPVI). Set to True unless relationships are not required.
Create Object Model File	Controls creation of a debugging HTML object model. Set to False because this can be time consuming and usually is unnecessary.

Property	Specify…
Delete Policy Timeout	The time interval, in minutes, before deleting an inactive element from the hierarchy. The default is 1440.
Discover Policy	The time interval, in minutes, between discovery sessions. The default is 1440.
Element Name Grouping Length	Controls element grouping. If set to zero, then no element grouping is performed and no lazy discovery is performed. If set to a non-zero value, then the elements under the four component types are grouped by the first n characters in their names. Example: If set to 3, then HOSTS `abcone`, `abctwo`, and `abcthree` are all grouped under the element `abc`. After setting a grouping length, do not change it, as this also changes the DName of all elements under the grouping.
Grouping Elements Discover Only When Needed	If the Element Name Grouping Length property is non-zero, then this controls whether the elements under the grouping are lazily discovered. Set to True if using grouping.
Hierarchy File	A file in the */OperationsCenter_install_path/*`database` directory that contains an XML description of the element hierarchy built below the adapter element. The default is `examples/FoundationHierarchy.xml`.
Script.onError	A script that executes if the adapter fails for any reason. The script can print the reason for the failure as a “msg”; for example: log.info(msg). All of the `Script.*` properties are optional.
Script.onInitialized	A script that executes when the adapter initializes.
Script.onStarted	A script that executes when the adapter starts, either manually or automatically when the Operations Center server starts.
Script.onStopped	A script that executes after manually stopping the adapter.
Stylesheet File	The stylesheet file in the */OperationsCenter_install_path/*`database` directory that applies to the HierarchyFile as a style markup to produce the final output.
Use Alarm Times For Condition Changes	The date/time stamp to use for all alarm data stored by the Operations Center Data Warehouse. If `true`, the alarm’s date/time stamp is used. If `false`, the date/time stamp of when the Operations Center server received the alarm is used. The default is `true`. Alarm history is stored based on the alarm time rather than alarm receipt time. Also, for SLA metric data based on alarm properties, the property values are recorded based on the alarm time instead of the alarm receipt time. Note that recording historical condition data for historical alarms is not supported.

B Documentation Updates

This section contains information on documentation content changes that were made in the *Adapter and Integration Guide* after the initial release of Operations Center 5.0. The changes are listed according to the date they were published.

If you need to know where a copy of the PDF documentation you are using is the most recent, the PDF document includes a publication date on the title page.

February 19, 2015

A video tutorial is available on YouTube for creating adapters. For step-by-step instructions and a link to the video tutorial, see Section 2.1, “Creating an Adapter,” on page 17.