

Server Configuration and Customization Guide

Operations Center 5.5

November 18, 2014

Legal Notices

THIS DOCUMENT AND THE SOFTWARE DESCRIBED IN THIS DOCUMENT ARE FURNISHED UNDER AND ARE SUBJECT TO THE TERMS OF A LICENSE AGREEMENT OR A NON-DISCLOSURE AGREEMENT. EXCEPT AS EXPRESSLY SET FORTH IN SUCH LICENSE AGREEMENT OR NON-DISCLOSURE AGREEMENT, NETIQ CORPORATION PROVIDES THIS DOCUMENT AND THE SOFTWARE DESCRIBED IN THIS DOCUMENT "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. SOME STATES DO NOT ALLOW DISCLAIMERS OF EXPRESS OR IMPLIED WARRANTIES IN CERTAIN TRANSACTIONS; THEREFORE, THIS STATEMENT MAY NOT APPLY TO YOU.

For purposes of clarity, any module, adapter or other similar material ("Module") is licensed under the terms and conditions of the End User License Agreement for the applicable version of the NetIQ product or software to which it relates or interoperates with, and by accessing, copying or using a Module you agree to be bound by such terms. If you do not agree to the terms of the End User License Agreement you are not authorized to use, access or copy a Module and you must destroy all copies of the Module and contact NetIQ for further instructions.

This document and the software described in this document may not be lent, sold, or given away without the prior written permission of NetIQ Corporation, except as otherwise permitted by law. Except as expressly set forth in such license agreement or non-disclosure agreement, no part of this document or the software described in this document may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, or otherwise, without the prior written consent of NetIQ Corporation. Some companies, names, and data in this document are used for illustration purposes and may not represent real companies, individuals, or data.

This document could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein. These changes may be incorporated in new editions of this document. NetIQ Corporation may make improvements in or changes to the software described in this document at any time.

U.S. Government Restricted Rights: If the software and documentation are being acquired by or on behalf of the U.S. Government or by a U.S. Government prime contractor or subcontractor (at any tier), in accordance with 48 C.F.R. 227.7202-4 (for Department of Defense (DOD) acquisitions) and 48 C.F.R. 2.101 and 12.212 (for non-DOD acquisitions), the government's rights in the software and documentation, including its rights to use, modify, reproduce, release, perform, display or disclose the software or documentation, will be subject in all respects to the commercial license rights and restrictions provided in the license agreement.

© 2014 NetIQ Corporation. All Rights Reserved.

For information about NetIQ trademarks, see https://www.netiq.com/company/legal/ (https://www.netiq.com/company/legal/).

All third-party trademarks are the property of their respective owners.

Contents

About This Guide 9

Part I Configuring the Operations Center Server 13

1 Overview of Post-Installation Configurations 15

2 Updating Server Settings Using the Configuration Manager 17

2.1 Accessing and Using Configuration Manager . . . 17
2.2 Understanding Configuration Settings . . . 18
2.2.1 Host Pane . . . 19
2.2.2 Daemon Pane . . . 20
2.2.3 Security Pane . . . 22
2.2.4 Web Server Pane . . . 25
2.2.5 NOC Server Pane . . . 26
2.2.6 Image Server Pane . . . 28
2.2.7 SQL Views Pane . . . 29
2.2.8 Remote Container Pane . . . 30
2.2.9 Event Manager Pane . . . 31
2.2.10 Event Manager Agent Pane . . . 33
2.2.11 Database Pane . . . 35
2.3 Setting Up Configuration File Backups . . . 36
2.4 Making Custom Changes . . . 37

3 Configuring Operations Center Start Conditions 39

3.1 Configuring mosdaemon . . . 39
3.1.1 Starting mosdaemon . . . 39
3.1.2 Configuring Access to mosdaemon . . . 41
3.1.3 Checking the mosdaemon Status . . . 41
3.2 Configuring an Auto-Restart . . . 41
3.2.1 Configuring the Windows Service . . . 42
3.2.2 Configuring Daemon Control . . . 43
3.2.3 Stopping mosmonitor from Automatically Restarting Operations Center . . . 44
3.3 Manually Starting Operations Center . . . 44
3.4 Manually Stopping Operations Center Components . . . 44
3.4.1 Immediately Stopping All of Operations Center . . . 45
3.4.2 Stopping Operations Center with a Delay . . . 45
3.4.3 Stopping Only a Particular Component . . . 45
3.5 Configuring Web Server Start and Stop . . . 45
3.5.1 Manually Starting or Stopping the Web Server . . . 46
3.5.2 Configuring Server Initialization and Messaging Settings . . . 46
3.6 Manually Starting or Stopping the Image Server . . . 47

4 Configuring Network Communication Settings 49

4.1 Configuring Server IP Addressing . . . 49
4.1.1 Restricting Access by IP Address . . . 49
4.1.2 Configuring NAT . . . 50
4.2 Configuring Ports . . . 51

4.2.1 Identifying Ports in Use . . . 52
4.2.2 Configuring a Operations Center Server Port Range . . . 52
4.2.3 Configuring the Operations Center Port Assignments . . . 54

5 Configuring Trace Logs 57

5.1 Understanding Trace Logs and Content . . . 57
5.2 Changing the Persistent Component Log Settings . . . 58
5.2.1 Trace Destination . . . 59
5.2.2 Trace Level . . . 61
5.3 Configuring Current Session Element Log Settings . . . 61
5.3.1 Selecting the Trace Level, then the Server Element . . . 61
5.3.2 Selecting the Server Element, then the Trace Level . . . 62

6 Configuring Java and Memory 63

6.1 Configuring the Java Virtual Machine . . . 63
6.1.1 About Java and Memory . . . 63
6.1.2 Understanding Windows Memory Restrictions . . . 64
6.1.3 Understanding Memory Allocation . . . 64
6.1.4 Understanding Memory Parameters . . . 65
6.1.5 Configuring Operations Center's Use of Memory . . . 66
6.1.6 Resolving Out of Memory Errors in Trace Logs . . . 67
6.2 Configuring the Java Runtime Environment . . . 68

7 Configuring and Administering the Database 69

7.1 Configuring Windows Servers for Single Sign On (SSO) . . . 70
7.2 Configuring the Database for Configuration Storage . . . 70
7.2.1 Configuring the Database . . . 71
7.2.2 Specifying a Configuration Storage Database . . . 72
7.2.3 Configuring Maintenance Options for the Configuration Storage Database . . . 74
7.3 Creating Database Definitions, Configuring the Event Data Store, and Connecting to External Databases . . . 75
7.3.1 Setting Up an External Database . . . 76
7.3.2 Creating and Editing a Database Definition . . . 86
7.3.3 Enabling a Definition . . . 88
7.3.4 Initializing a Definition . . . 88
7.3.5 Disabling and Deleting a Definition . . . 89
7.4 Viewing Database Status and Statistics . . . 89
7.5 Configuring the Service Warehouse . . . 90
7.5.1 Sizing the Service Warehouse . . . 91
7.5.2 Configuring the Service Warehouse . . . 95
7.5.3 Customizing Data Collection Settings for Alarms and Performance Metrics . . . 97
7.5.4 Enabling the Service Warehouse Backup Repository . . . 99
7.5.5 Enabling and Disabling the Service Warehouse . . . 101
7.5.6 Auditing Service Warehouse Events . . . 101
7.5.7 Viewing Service Warehouse Status & Statistics . . . 102
7.6 About Operations Center Embedded Databases . . . 103
7.6.1 The Dashboard . . . 103
7.6.2 SQL Views . . . 103

8 Managing Configurations 105

8.1 Using the Configuration Explorer . . . 105
8.1.1 Accessing Configuration Explorer . . . 105
8.1.2 Understanding the Menu Options . . . 106

8.2 Setting the Configuration Default . . . 108
8.3 Creating Configurations . . . 108
8.4 Modifying Configurations . . . 110
8.4.1 Changing Configuration Storage Type . . . 110
8.4.2 Adding an Element or Property . . . 110
8.4.3 Renaming Elements . . . 111
8.4.4 Removing Children . . . 111
8.4.5 Reinitializing and Deleting . . . 111
8.5 Backing Up, Copying, and Restoring Configurations . . . 112
8.5.1 Using the Configuration Storage Database File . . . 112
8.5.2 Using the Configuration Explorer . . . 113
8.5.3 Using the Exportcfg and Importcfg Utilities . . . 114
8.5.4 Using Import/Export from the Operations Center Console . . . 116
8.5.5 Using NOC Script to Import and Export Configuration XML Files . . . 119
8.5.6 Using NOC Script to Backup the Configuration to an ODB Database File . . . 124

9 Administering the Operations Center Server 125

9.1 Viewing Operations Center Server System Information . . . 125
9.1.1 Viewing Statuses for the Operations Center Server . . . 126
9.1.2 Using the mosstatus Command . . . 127
9.2 Using Server Scripts . . . 128
9.3 Using Jobs . . . 128
9.3.1 Accessing Jobs . . . 129
9.3.2 Scheduling Jobs . . . 129
9.3.3 Running Jobs . . . 134
9.3.4 Viewing the Status of Jobs . . . 134
9.3.5 Enabling and Disabling Jobs . . . 135
9.3.6 Stopping Jobs . . . 136
9.3.7 Changing Job Definitions . . . 137
9.4 Using the Web Server . . . 137
9.5 Using the Image Server . . . 137

Part II Customizing the Implementation 141

10 Enabling and Disabling Console Functionality 143

10.1 Customizing the Console to Restore Windows . . . 143
10.2 Overriding the Maximum Elements Limit for the Network View . . . 143
10.3 Disabling the Network and Layout Views . . . 144
10.4 Configuring Element Find Features . . . 144
10.4.1 Enabling Metamodel Classes or Properties as Search Criteria . . . 145
10.4.2 Disabling the Find Feature . . . 145
10.5 Enabling Chat . . . 146

11 Customizing Monitored Elements and Alarms 147

11.1 Customizing Icon Display . . . 148
11.2 Determining an Element's Class Name . . . 150
11.3 Displaying Element Availability Time . . . 150
11.4 Modifying Element and Alarm Menus . . . 151
11.4.1 Creating New Operations Through the Operations Center Console . . . 152
11.4.2 Adding New Menu Items in the Operations.ini File . . . 154
11.4.3 Adding Colored Menu Operations . . . 155
11.4.4 Optimizing Alarm Selection . . . 155
11.5 Filtering Alarm Columns . . . 156

11.6 Configuring Suppression and Acknowledgement . . . 157
11.6.1 Configuring Suppression/Acknowledgement Functions . . . 158
11.6.2 Changing the Permission Level for Suppression . . . 158
11.6.3 Configuring Reacknowledgement on Server Restart . . . 159
11.6.4 Configuring Suppression/Acknowledgement to Apply to a Single Element . . . 159
11.6.5 Enabling Suppression after Correcting Database Problems . . . 159
11.6.6 Changing the Acknowledge and Suppress Menu Names . . . 160
11.6.7 Using the Suppression and Acknowledgement Options . . . 160
11.7 Controlling Display of Alarms . . . 163
11.7.1 Configuring Maximum Alarms . . . 164
11.7.2 Limiting the Roll Up of Alarms to Root Elements . . . 164
11.7.3 Displaying a Message for Paused Alarms . . . 165
11.7.4 Resuming Alarms after Pause . . . 165
11.7.5 Pause Alarms on Right-Click . . . 165
11.8 Filtering Service Model Alarms . . . 166
11.8.1 Creating Alarm Filters . . . 167
11.8.2 Editing Alarm Filters . . . 169
11.8.3 Creating Filter Groups . . . 170
11.8.4 Applying Filters to Service Models . . . 171
11.8.5 Applying Filters to Element Classes . . . 172
11.8.6 Clearing Filters from Elements . . . 173
11.8.7 Giving Users View Permissions to Filters . . . 173
11.8.8 Deleting Filters . . . 173
11.8.9 Assigning Filters Using Scripts . . . 174
11.9 Managing Administration Elements on Remote Servers . . . 174

12 Capturing Alarm and Performance History 177

12.1 Alarm History Overview . . . 177
12.2 Using Profiles and Expressions to Capture Alarm History . . . 178
12.2.1 Creating Profiles . . . 178
12.2.2 Creating Expressions . . . 184
12.2.3 Starting and Stopping Profiles . . . 189
12.2.4 Monitoring Profile Data Collection . . . 191
12.3 Assigning Profiles at the Element Level . . . 192
12.4 Viewing Alarm History . . . 192
12.5 Monitoring Alarm History Collection . . . 193
12.5.1 Changing Query Limit Values . . . 194
12.5.2 Changing Database Query Time Limits . . . 194
12.5.3 Stopping and Starting Data Collection . . . 194
12.5.4 Viewing Data Warehouse State and Statistics . . . 194
12.6 Customizing Default Performance Chart Settings . . . 195

13 Time Categories, Calendars, and Schedules 197

13.1 Maintaining Time Categories . . . 198
13.2 Creating and Maintaining Calendars . . . 200
13.2.1 Creating Calendars . . . 201
13.2.2 Specifying Time Definitions . . . 202
13.2.3 Using Existing Calendars to Create New Calendars . . . 203
13.2.4 Navigating the Calendar Visualization Section . . . 204
13.2.5 Assigning Undesignated Times . . . 205
13.2.6 Editing Calendars . . . 206
13.2.7 Setting Blackout Calendars on Elements . . . 208
13.3 Creating and Editing Schedules . . . 209
13.3.1 Creating a Schedule . . . 209
13.3.2 Editing a Schedule . . . 211
13.3.3 Deleting a Schedule . . . 212

14 Defining and Managing Automation Events 213

14.1 Understanding Client-Side and Server Side Automations . . . 214
14.2 Defining Automation Events . . . 216
14.3 Monitoring and Managing Automation Events . . . 218
14.3.1 Monitoring Your Client-Side Automation Events . . . 219
14.3.2 Viewing Statistics about Server Side Automation Tasks . . . 220
14.3.3 Viewing the Server-Side Automations Queue . . . 221
14.3.4 Clearing the Server-Side Automations Queue . . . 221
14.3.5 Canceling the Current Server-Side Automation Task . . . 222
14.3.6 Editing Automation Events . . . 222
14.4 Defining Automation Events Directly on Elements . . . 223
14.5 Monitoring and Managing Automation Events for an Element . . . 224
14.5.1 Monitoring Automation Events for an Element . . . 224
14.5.2 Modifying Automations for Elements . . . 224
14.6 Using Automation Filters . . . 226
14.6.1 Understanding the Default Filters . . . 226
14.6.2 Filtering Alarm Events . . . 228
14.6.3 Defining a New Filter . . . 229
14.6.4 Modifying a Filter . . . 230
14.6.5 Deleting a Filter . . . 230
14.7 Using Automation Actions . . . 231
14.7.1 Understanding Action Parameters . . . 231
14.7.2 Using the Open Alarm Pop-up Dialog Box Action . . . 233
14.7.3 Defining a New Action . . . 233
14.7.4 Modifying an Automation Action . . . 234
14.7.5 Deleting an Automation Action . . . 235

15 Using Algorithms to Calculate Element State 237

15.1 Understanding Algorithms . . . 237
15.1.1 Default Algorithm Types . . . 238
15.1.2 Custom Algorithm Examples . . . 238
15.2 Setting an Algorithm for an Element . . . 239
15.3 Verifying and Troubleshooting Algorithms Using the Algorithm Tracer . . . 240
15.3.1 Tracing and Validating the Algorithm Calculation . . . 241
15.3.2 Testing an Algorithm's Effect on Parent Condition . . . 242
15.4 Modifying the Algorithm Library . . . 242
15.5 Algorithm XML Tags Reference . . . 243
15.5.1 Understanding the Algorithms.xml File . . . 243
15.5.2 Understanding the Top-Level Tags . . . 244
15.5.3 Understanding timebasedbranch and timebasedsplit . . . 250
15.5.4 Understanding Parameter Evaluation . . . 251

A The Operations Center XML Editor 253

A.1 Understanding and Accessing the XML Editor . . . 253
A.2 Managing XML Files . . . 254
A.2.1 Understanding the XML Editor Toolbar . . . 255
A.2.2 Using the Document Type Definition (DTD) . . . 255
A.3 Adding Elements . . . 257
A.3.1 Adding a Tag as a Subelement . . . 257
A.3.2 Placing a New Element . . . 258
A.3.3 Adding Comments and Processing Instructions . . . 258
A.4 Adding and Editing Attributes . . . 259
A.5 Moving Tags . . . 259
A.5.1 Moving a Tag Element . . . 260
A.5.2 Cutting, Copying, and Pasting Tags . . . 260

A.6 Deleting Tags 260

About This Guide

The *Server Configuration Guide* provides post-installation instructions for configuring and customizing the Operations Center server and console.

Part I, "Configuring the Operations Center Server," on page 13

- Chapter 1, "Overview of Post-Installation Configurations," on page 15

  Provides an introduction to this guide.
- Chapter 2, "Updating Server Settings Using the Configuration Manager," on page 17

  Explains how this tool is used for configuration.
- These sections provide the configuration tasks:

  Chapter 3, "Configuring Operations Center Start Conditions," on page 39

  Chapter 4, "Configuring Network Communication Settings," on page 49

  Chapter 5, "Configuring Trace Logs," on page 57

  Chapter 6, "Configuring Java and Memory," on page 63

  Chapter 3, "Configuring Operations Center Start Conditions," on page 39
- Chapter 7, "Configuring and Administering the Database," on page 69

  Describes the databases that need to be created, configured, and administered (including Configuration Storage).
- Chapter 8, "Managing Configurations," on page 105

  Explains how configurations can be copied, shared, and moved.
- Chapter 9, "Administering the Operations Center Server," on page 125

  Explains how to administer the Operations Center server (including running jobs), the Web server, and the Image server.

Part II, "Customizing the Implementation," on page 141

- Chapter 10, "Enabling and Disabling Console Functionality," on page 143
- Chapter 11, "Customizing Monitored Elements and Alarms," on page 147
- Chapter 12, "Capturing Alarm and Performance History," on page 177
- Chapter 13, "Time Categories, Calendars, and Schedules," on page 197
- Chapter 14, "Defining and Managing Automation Events," on page 213
- Chapter 15, "Using Algorithms to Calculate Element State," on page 237

Appendix A, "The Operations Center XML Editor," on page 253

Audience

This guide is intended for the Operations Center administrator who sets up, uses, and maintains Operations Center servers.

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 & Documentation Updates

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.

Additional Resources

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

- NetIQ User Community (https://www.netiq.com/communities/): A Web-based community with a variety of discussion topics.
- NetIQ Support Knowledgebase (https://www.netiq.com/support/kb/product.php?id=SG_XOPERATIONSCENTER_1_2): A collection of in-depth technical articles.
- NetIQ Support Forums (https://forums.netiq.com/forumdisplay.php?26-Operations-Center): 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 (https://www.netiq.com/Support/process.asp#_Maintenance_Programs_and).

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 (support@netiq.com)
- Submit a Service Request: http://support.NetIQ.com/contact/ (http://support.novell.com/contact/)

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 pathname can be written with a backslash for some platforms or a forward slash for other platforms, the pathname 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.

Configuring the Operations Center Server

After installing Operations Center, you need to make various initial configurations to Operations Center server. You might also need to tweak configuration settings further for better performance or for certain requirements that your environment demands.

If you have just installed Operations Center, start with Overview of Post-Installation Configurations (page 15) for an introduction to the configurations you'll need to set up for your implementation.

This section covers the following post-installation configuration topics:

- Chapter 1, "Overview of Post-Installation Configurations," on page 15
- Chapter 2, "Updating Server Settings Using the Configuration Manager," on page 17
- Chapter 3, "Configuring Operations Center Start Conditions," on page 39
- Chapter 4, "Configuring Network Communication Settings," on page 49
- Chapter 5, "Configuring Trace Logs," on page 57
- Chapter 6, "Configuring Java and Memory," on page 63
- Chapter 7, "Configuring and Administering the Database," on page 69
- Chapter 8, "Managing Configurations," on page 105
- Chapter 9, "Administering the Operations Center Server," on page 125

1 Overview of Post-Installation Configurations

After installing NetIQ® Operations Center, it is necessary to perform the following actions and configurations:

1. If you are installing on a cluster, do all of the following steps after completing installations on all servers. For more information on installing and configuring Operations Center servers in a clustered environment, see "Implementing a High Availability Solution" in the *Operations Center 5.5 Server Installation Guide*.
2. Run the Configuration Manager, an Operations Center utility used to set configuration settings for the Operations Center servers and for Configuration Storage. Specify server settings for networking, trace logs, java, memory, and more. For instructions on using the Configuration Manager, see Chapter 2, "Updating Server Settings Using the Configuration Manager," on page 17.

   At any time, specific configurations can be created and copied for ease in moving Operations Center (for example from a development or test environment to production) by using the Configuration Explorer utility. Configurations are stored in one of these data stores called Configuration Storage. For instructions on using the Configuration Explorer, see Chapter 8, "Managing Configurations," on page 105
3. Configure the Operations Center start options. Refer to Configuring Operations Center Start Conditions (page 39) for details.
4. Perform further customizations as required for networking, trace logs, and Java and memory settings. Refer to the following sections for instructions:
   - Configuring Network Communication Settings (page 49)

     Operations Center is installed using default ports, but do consider the ports currently used and how they fit within your environment.
   - Configuring Trace Logs (page 57)

     Configure trace logs for the Operations Center server and other components. Define settings applicable for the current session and those that will persist for each session.
   - Configuring Java and Memory (page 63)

     The Java Virtual Machine is configured with parameters for memory allocation and installs with these default values. Before adjusting the memory, it is important to understand how memory is allocated.
5. If you plan to use any external databases with Operations Center, you'll need to create the database instance. Both the Operations Center data store and any external databases require a database definition to be created and managed in the Operations Center console. Refer to the following sections for instructions:
   - Configuring the Database for Configuration Storage (page 70)

The Operations Center data store is an embedded Object ODB that stores Operations Center configuration data, version tracking, and to control connections to all databases configured for use with Operations Center. Configure an external database to handle this data repository.

- Setting Up an External Database (page 76)

  Run scripts to create and configure the database instance before configuring Operations Center to connect with the database.

- Creating and Editing a Database Definition (page 86)

  Create a database definition for each database connection.

- Configuring the Service Warehouse (page 90)

  The Service Warehouse stores alarm history, alarm comments, historical performance and service level data. Even if you do not plan to use SLAs and Service Level Manager, create and define a Service Warehouse for alarm history data.

6. Set up server security settings.

   For information about security in Operations Center, see the *Operations Center 5.5 Security Management Guide*.

7. Customize the implementation for users and administrators by configuring how elements and alarms are displayed in the Operations Center console, how alarm and performance history is captured, when alerts are issued after network events, and how element states are calculated. For information and instructions, see Part II, “Customizing the Implementation,” on page 141.

2 Updating Server Settings Using the Configuration Manager

Configuration Manager is a utility in Operations Center through which you set configuration settings for the components of Operations Center including the Operations Center server, Web server, Image server, and databases, as well as some other Operations Center applications.

Configuration Manager can be accessed during the installation process or after Operations Center has been installed. For more information about how to install Operations Center, see the *Operations Center 5.5 Server Installation Guide*.

The settings in Configuration Manager are described briefly in this section. Be sure to review the other relevant sections of this guide as well as other guides before changing any settings.

Most configuration settings can be configured by using the Configuration Manager and are stored in the `formula.properties` file. The `formula.properties` file should never be edited directly. Custom changes not configurable through the Configuration Manager can be made to the configuration file by using the `Formula.custom.properties` file, which is never overwritten by the Configuration Manager.

WARNING: Any configuration changes made outside the Configuration Manager might be overwritten the next time the Configuration Manager is run.

After making changes in Configuration Manager, you might want to consider creating backup copies of configuration files in case changes are made that are determined to be unacceptable.

Review the following sections to understand configuration settings and how to update them:

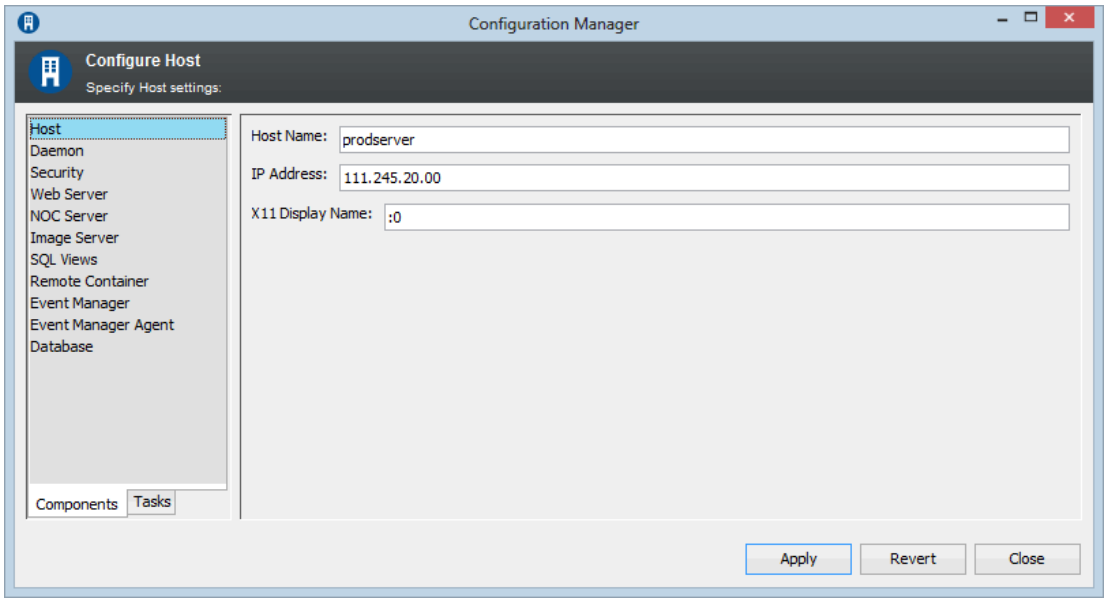
- Section 2.1, “Accessing and Using Configuration Manager,” on page 17
- Section 2.2, “Understanding Configuration Settings,” on page 18
- Section 2.3, “Setting Up Configuration File Backups,” on page 36
- Section 2.4, “Making Custom Changes,” on page 37

2.1 Accessing and Using Configuration Manager

You can opt to open Configuration Manager immediately after installing Operations Center.

1 To open Configuration Manager, do one of the following:
   - **Windows:** You have two options:
     - Click *Start* > *All Programs* > *NetIQ Operations Center* > *Configure NetIQ Operations Center*.
     - At the DOS prompt, run `customizer` from the `\OperationsCenter_install_path\bin` directory.
   - **UNIX:** At a command prompt, run `Customizer` from the `/OperationsCenter_install_path/bin` directory.

The following window opens:



2 To use Configuration Manager (some basic actions):

- Click either the *Components* or *Tasks* tab, then select a pane from the list.
  - **Components:** This tab has all of the available settings, separated according to component.

    See Section 2.2, "Understanding Configuration Settings," on page 18 for descriptions and instructions on each of the component-related settings.
  - **Tasks:** This tab has options for *Networking*, *Java Runtime*, and *Logging*, subdivided into these three categories.
- To display a tooltip description of a setting, hover the mouse over the field or drop-down list.
- If you installed only the Event Manager, configure only the options on the *Event Manager* pane (located on the *Components* tab).

3 After making changes, click *Apply* to save the changes; or click *Revert* to revert to previously saved values.

4 Continue with Section 2.3, "Setting Up Configuration File Backups," on page 36.

2.2 Understanding Configuration Settings

When a configuration setting is available on both the *Components* tab and *Tasks* tab, it can be changed in either location.

The settings available through the *Tasks* tab are a smaller subset of those offered in *Components* tab, and are only the settings related to *Networking*, *Java Runtime*, and *Logging*. If you are looking for more information about a setting while using the *Tasks* tab, refer to the appropriate component topic in this section.

The figures in following sections display the settings available on the different panes for the *Components* tab. The table that follows each figure briefly describes each setting with a link (or reference) to additional information:

- Section 2.2.1, “Host Pane,” on page 19
- Section 2.2.2, “Daemon Pane,” on page 20
- Section 2.2.3, “Security Pane,” on page 22
- Section 2.2.4, “Web Server Pane,” on page 25
- Section 2.2.5, “NOC Server Pane,” on page 26
- Section 2.2.6, “Image Server Pane,” on page 28
- Section 2.2.7, “SQL Views Pane,” on page 29
- Section 2.2.8, “Remote Container Pane,” on page 30
- Section 2.2.9, “Event Manager Pane,” on page 31
- Section 2.2.10, “Event Manager Agent Pane,” on page 33
- Section 2.2.11, “Database Pane,” on page 35

2.2.1 Host Pane

The section describes Configuration Manager settings for the Operations Center host server.

Figure 2-1 shows the Host pane that contains configuration options necessary for the connection with the Operations Center host server. Table 2-1 on page 20 provides information about configuring these options.

Figure 2-1 *Configuration Manager Host Pane on the Components Tab (on Windows)*

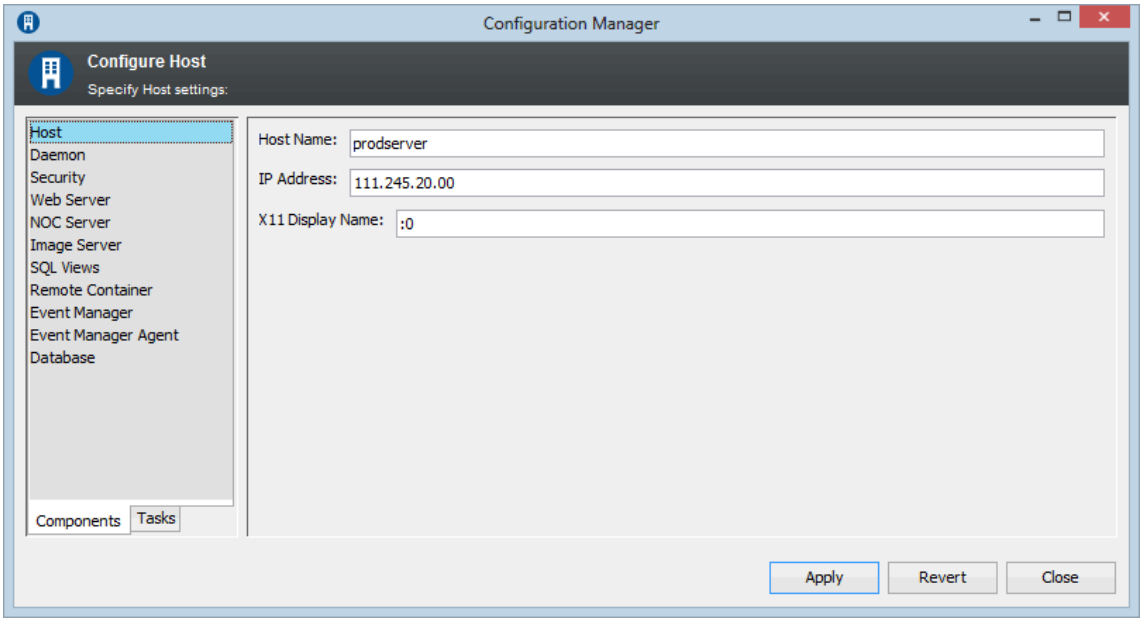


Table 2-1 *Configuration Manager Host Pane Settings*

Setting	Windows Default	UNIX Default	Description
Host Name	Hostname of the Primary Network Interface	The local server hostname	TCP/IP hostname of the server. See Chapter 4, “Configuring Network Communication Settings,” on page 49.
IP Address	Primary Network Interface	The local server IP address	The IP address or the TCP/IP hostname of the server. By default, the IP address for the Primary Network Interface is prefilled. See Chapter 4, “Configuring Network Communication Settings,” on page 49.
X11 Display Name	Not applicable to Windows	`:1`	(Unix Only) The display used by the virtual framebuffer. Optionally specify host name and screen name: *`host:display_name:screen_name`* See *Configuring Display for Images on UNIX* in the *Operations Center 5.5 Server Installation Guide*.

2.2.2 Daemon Pane

The section describes Configuration Manager settings for the Operations Center server daemon process.

Figure 2-2 shows the Daemon pane that contains configuration options for the daemon process. Table 2-2 on page 21 provides information about configuring these options.

Figure 2-2 *Configuration Manager Daemon Pane on the Components Tab*

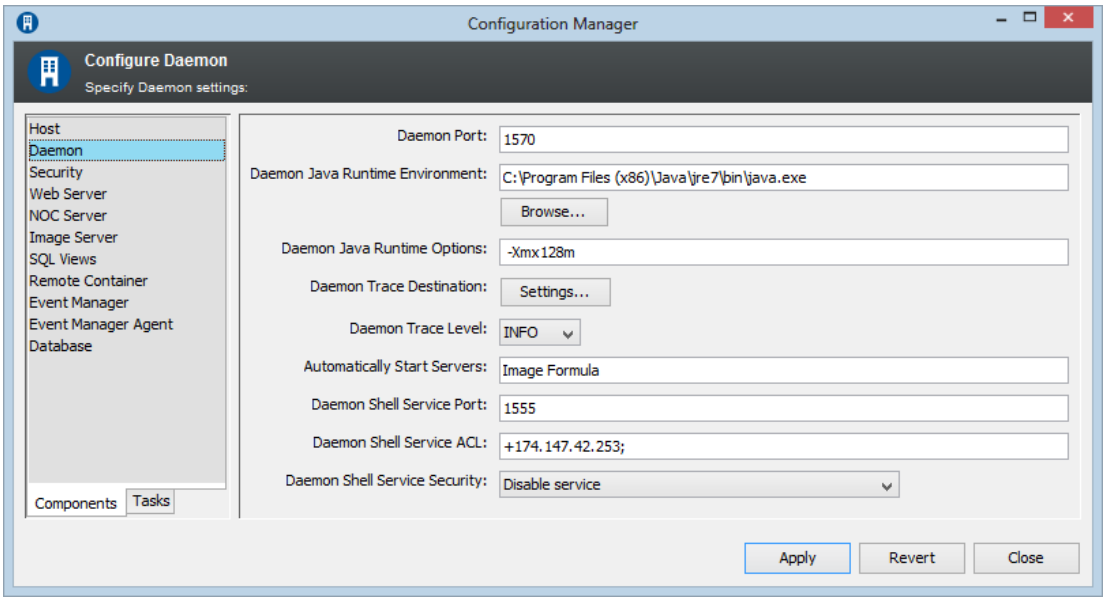


Table 2-2 *Configuration Manager Daemon Pane Settings*

Setting	Windows Default	UNIX Default	Description
Daemon Port	1570	1570	Port number for the Operations Center daemon.
Daemon Java Runtime Environment	`C:\`*`OperationsCenter_install_path`*`\jre\bin\java.exe`	*`/OperationsCenter_install_path`*`/jre/bin/java.exe`	Executable for the Java Runtime Environment for the daemon for Operations Center. Click *Browse* to navigate to the location of the JRE. See Chapter 6, “Configuring Java and Memory,” on page 63.
Daemon Java Runtime Options	`-server -Xmx128m`	`-server -Xmx256m`	Options for running the VM for the Operations Center background daemon. Usually does not need to be changed. See Chapter 6, “Configuring Java and Memory,” on page 63.
Daemon Trace Destination	N/A	N/A	The settings for the daemon trace logs. See Chapter 5, “Configuring Trace Logs,” on page 57.
Daemon Trace Level	INFO	INFO	Controls how much information is passed to the daemon trace logs. See Chapter 5, “Configuring Trace Logs,” on page 57.
Automatically Start Servers	Database Image Formula	N/A	A list of all servers that should start automatically when the mosdaemon starts. See Chapter 3, “Configuring Operations Center Start Conditions,” on page 39.
Daemon Shell Service Port	1555	1555	The specified port to listen for the daemon shell service.
Daemon Shell Service ACL	*`+IP address for localhost;`*	*`+IP address for localhost;`*	The clients running on the local host that can access the daemon shell service. To configure this setting, precede each hostname or IP address that is allolwed access with a plus sign (+), and separate the entries with a semicolon (;). For example: `+devtower10;+qasun1;`

Setting	Windows Default	UNIX Default	Description
Daemon Shell Service Security	Disable Service	Disable Service	Enables the daemon shell service, where by using and RMI registry in the daemon, processes can be forked to avoid generating a large swap footprint. Select one of the following options: ◆ Unsecured communication ◆ Secured communication using SSL ◆ Secured Communication using SSL and Client Certificates Default is `Disable service`. For more information about communication settings, see the "Communications Security" in the *Operations Center 5.5 Security Management Guide*.

2.2.3 Security Pane

The section describes Configuration Manager settings for secure communication and security features.

Figure 2-3 shows the Security pane that contains configuration options for secure communications between client and server, the Dashboard, and Web Services. Table 2-3 on page 23 provides information about configuring these options.

Figure 2-3 *Configuration Manager Security Pane on the Components Tab*

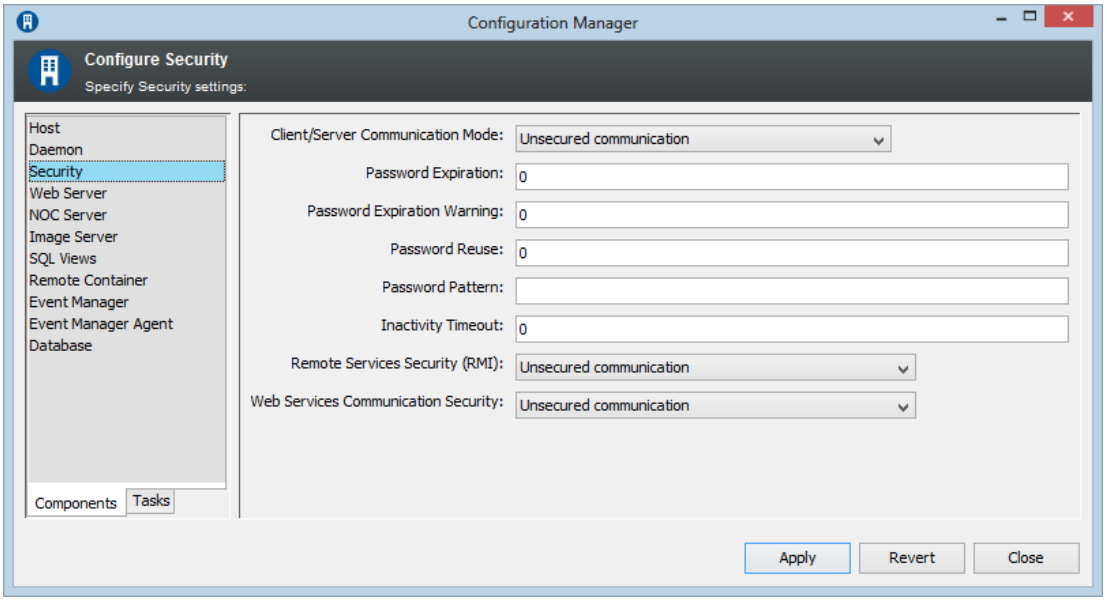


Table 2-3 *Configuration Manager Security Pane Settings*

Setting	Windows Default	UNIX Default	Description
Client/Server Communication Mode	Unsecured Communication	Unsecured Communication	Select from the following: ◆ **Unsecured Communication:** The server only accepts access via Hypertext Transfer Protocol (HTTP) communications protocols, and does not use SSL to encrypt these data streams. ◆ **Secured Communication using SSL:** The server only accepts access via HTTPS communications protocols, using the Secure Sockets Layer to encrypt these data streams. ◆ **Support Both Unsecured and Secured Communication:** Operations Center supports both secured and unsecured access. See Chapter 4, "Configuring Network Communication Settings," on page 49. For more information, see "Communications Security" in the *Operations Center 5.5 Security Management Guide*.
Password Expiration	0	0	The number of days a password can be used. For example, enter 30 to require users to change their passwords every 30 days.
Password Expiration Warning	0	0	The number of days prior to password expiration to issue a warning for users to change their passwords. Enter `0` to disable.
Password Reuse	0	0	Determines the number of previous passwords that can be reused. Specify `0` to use the same password indefinitely. Specify a positive number (n) to indicate that new passwords cannot be the same as those of the last n passwords. For example, if you enter 2, a user cannot specify the same password that was used the last two times. Specify a negative number (-n) to indicate that new passwords cannot be the same as those used for the past n days. The number of days specified must be equal to or less than the number of days entered for Password Expiration. For example, specify `-18` to prevent users from reusing passwords that were used during the past 18 days.

Setting	Windows Default	UNIX Default	Description
Password Pattern	0 or blank	0 or blank	Select one of the following to specify the format of the password: ◆ Enter `0` or leave it blank to specify no restriction on the password format. ◆ Enter `.*[0-9]+.*` to specify that the password might contain a combination of letters, digits, and special characters; however, it must contain at least a numeric character from 0 to 9. ◆ Specify a regular expression pattern that must be matched.
Inactivity Timeout	0 or blank	0 or blank	The time frame during which user sessions can remain inactive before they are required to log in again. 0 or blank specifies no timeout. A positive number specifies the length of time in minutes that the user can remain inactive. After the inactivity timeout occurs, users need to log on again and restart the Operations Center console.
Remote Services Security (RMI)	Unsecured communication	Unsecured communication	Three options for setting the RMI security level for communications between the Operations Center server and the dashboard, and between the Operations Center server and CMS: ◆ Unsecured communication ◆ Secured communication using SSL ◆ Secured communication using SSL and Client Certificates For more information about the dashboard, see the *Operations Center 5.5 Dashboard Guide*. For more information about CMS, see the *Operations Center 5.5 Configuration Management System (CMS) Guide*.
Web Services Communication Security	Unsecured communication	Unsecured communication	Three options for communication between third-party applications and the Operations Center Web Services Application Programmer Interface (WSAPI): ◆ Unsecured communication ◆ Secured communication using SSL ◆ Secured communication using SSL and Client Certificates For more information, see the *Operations Center 5.5 Web Services Guide*.

2.2.4 Web Server Pane

The section describes Configuration Manager settings for the Web Server.

Figure 2-4 shows the Web Server pane that contains configuration options necessary for communication with the Web server. Table 2-4 on page 25 provides information about configuring these options.

Figure 2-4 *Configuration Manager Web Server Pane on the Components Tab*

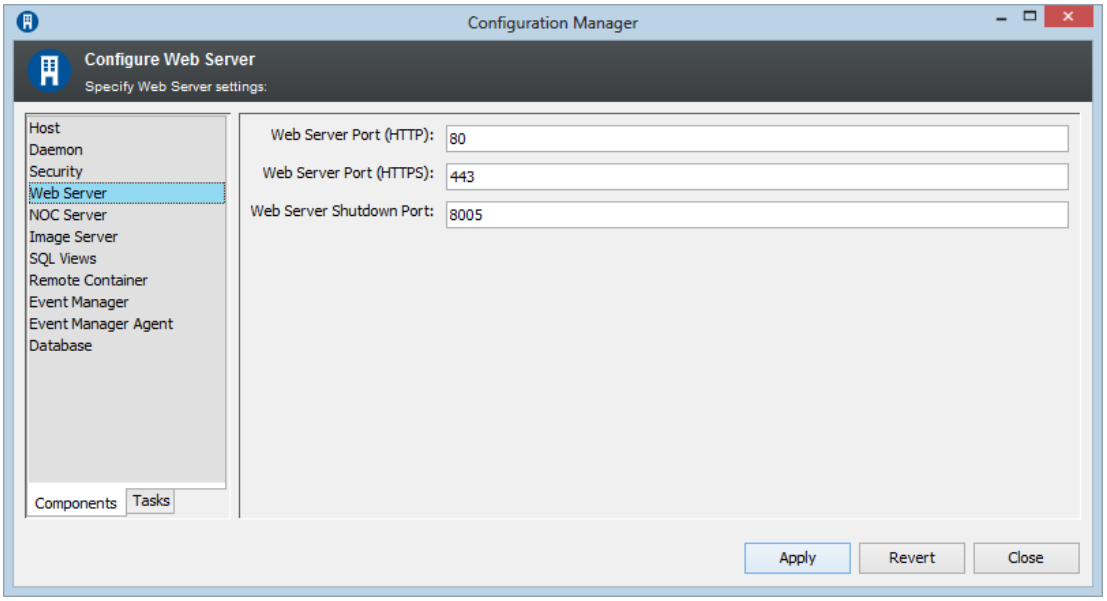


Table 2-4 *Configuration Manager Web Server Pane Settings*

Setting	Windows Default	UNIX Default	Description
Web Server Port (HTTP)	80	8080	Unsecured TCP/IP port number for Web server access. If the default port is already in use, change the port number. See Chapter 4, “Configuring Network Communication Settings,” on page 49.
Web Server Port (HTTPS)	443	443	Secure TCP/IP port number for Web server access. If the default port is already in use, change the port number. See Chapter 4, “Configuring Network Communication Settings,” on page 49.
Web Server Shutdown Port	8005	8005	The specified port opens a socket connection to listen for a shutdown command for the Web server. See Chapter 4, “Configuring Network Communication Settings,” on page 49.

2.2.5 NOC Server Pane

The section describes Configuration Manager settings for the Operations Center Server.

Figure 2-5 shows the NOC Server pane that contains configuration options for the server's runtime environment and options, log files, configuration storage, and service ports. Table 2-5 on page 26 provides information about configuring these options.

Figure 2-5 *Configuration Manager Server Pane on the Components Tab*

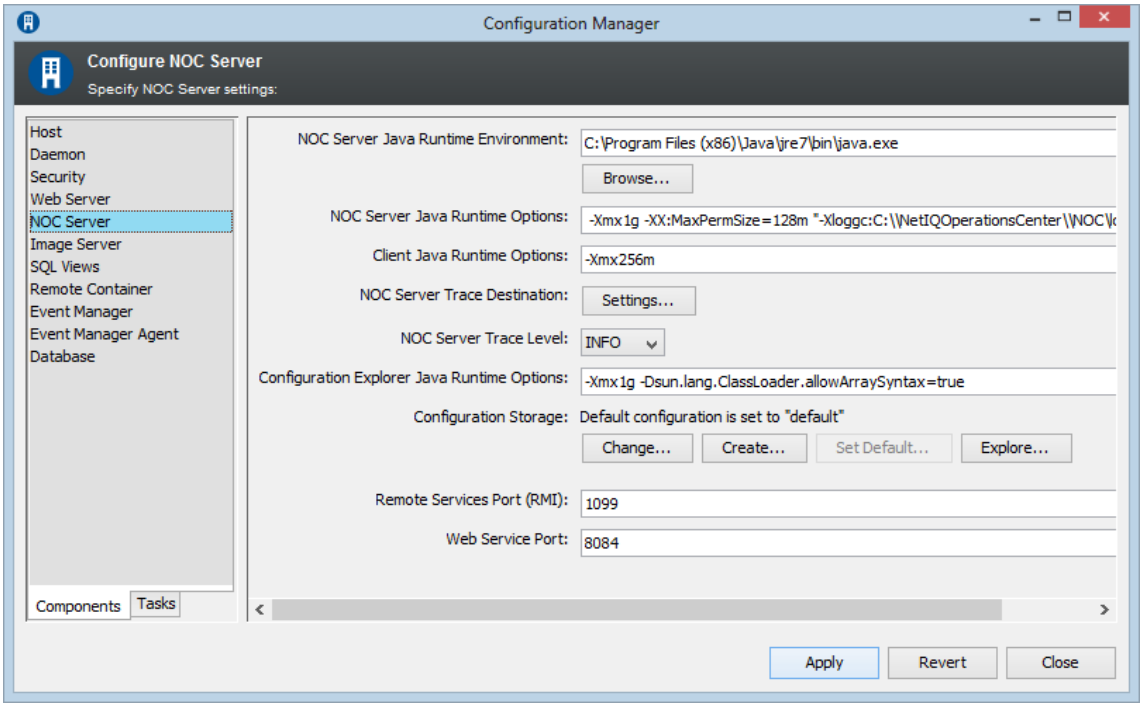


Table 2-5 *Configuration Manager Server Pane Settings*

Setting	Windows Default	UNIX Default	Description
NOC Server Java Runtime Environment	`c:\Program Files\`*`OperationsCenter_install_path`*`\jre\bin\java.exe`	`/`*`OperationsCenter_install_path`*`/jre/bin/jre`	Executable for the Java Runtime Environment (JRE) on your system. Click *Browse* to navigate to the location of the JRE program. Before changing this value, see Chapter 6, "Configuring Java and Memory," on page 63. For supported JREs, see the *Operations Center 5.5 Getting Started Guide*. For JRE installation information, see the *Operations Center 5.5 Server Installation Guide*.

Setting	Windows Default	UNIX Default	Description
NOC Server Java Runtime Options	-server -Xmx1g -XX:MaxPermSize=128m "-Xloggc:C:*OperationsCenter_install_path*\logs\fsgc.log"	-server -Xmx1g -XX:MaxPermSize=128m "-Xloggc:C:/*OperationsCenter_install_path*/logs/fsgc.log"	Options for running the Java Virtual Machine (JVM), which does not usually need to be changed. On an IBM JVM, change the `-Xloggc` parameter for verbose Garbage Collection logging to: `-Xverbosegclog:`*`OperationsCenter_install_path`*`/logs/fsgc.log`. If running in a 64-bit JVM, additional memory must be allocated. We recommend increasing `-XX:MaxPermSize` to `256m`. See Chapter 6, “Configuring Java and Memory,” on page 63.
Client Java Runtime Options	-Xmx256m	-server -Xmx512m -Xloggc:D:/formula3-2/logs/fsgc.log	Options for running the JVM for the Operations Center console. Subject to Java Web Start VM option restrictions. See Chapter 6, “Configuring Java and Memory,” on page 63.
Server Trace Destination	N/A	N/A	The settings for the Operations Center server trace logs. See Chapter 5, “Configuring Trace Logs,” on page 57.
NOC Server Trace Level	INFO	INFO	Controls how much information is passed to the server trace logs. It is recommended to leave the default setting at start up. See Chapter 5, “Configuring Trace Logs,” on page 57.
Configuration Explorer Java Runtime Options	-Xmx1g	-Xmx1g	Sets memory for the Configuration Explorer utility. See Chapter 6, “Configuring Java and Memory,” on page 63.
Configuration Storage	N/A	N/A	Each Operations Center server must have a Configuring the Database for Configuration Storage database. A default database is provided.
Remote Services Port (RMI)	1099	1099	Port number for the Remote Method Invocation (RMI) registry used by the dashboard. See Section 4.2, “Configuring Ports,” on page 51. This value must also be configured in the dashboards. For more information, see the *Operations Center 5.5 Dashboard Guide*.

Setting	Windows Default	UNIX Default	Description
Web Services Port	8084	8084	Port number for third-party applications to access the Operations Center server using the Operations Center Web Services Application Programmer Interface (WSAPI). See Section 4.2, “Configuring Ports,” on page 51. For more information, see the *Operations Center 5.5 Web Services Guide*.

2.2.6 Image Server Pane

The section describes Configuration Manager settings for the Image Server.

Figure 2-6 on page 28 shows the Image Server pane that contains configuration options for the image server ports, log files, and runtime options. Table 2-6 on page 28 provides information about configuring these options.

Figure 2-6 *Configuration Manager Image Server Pane on the Components Tab*

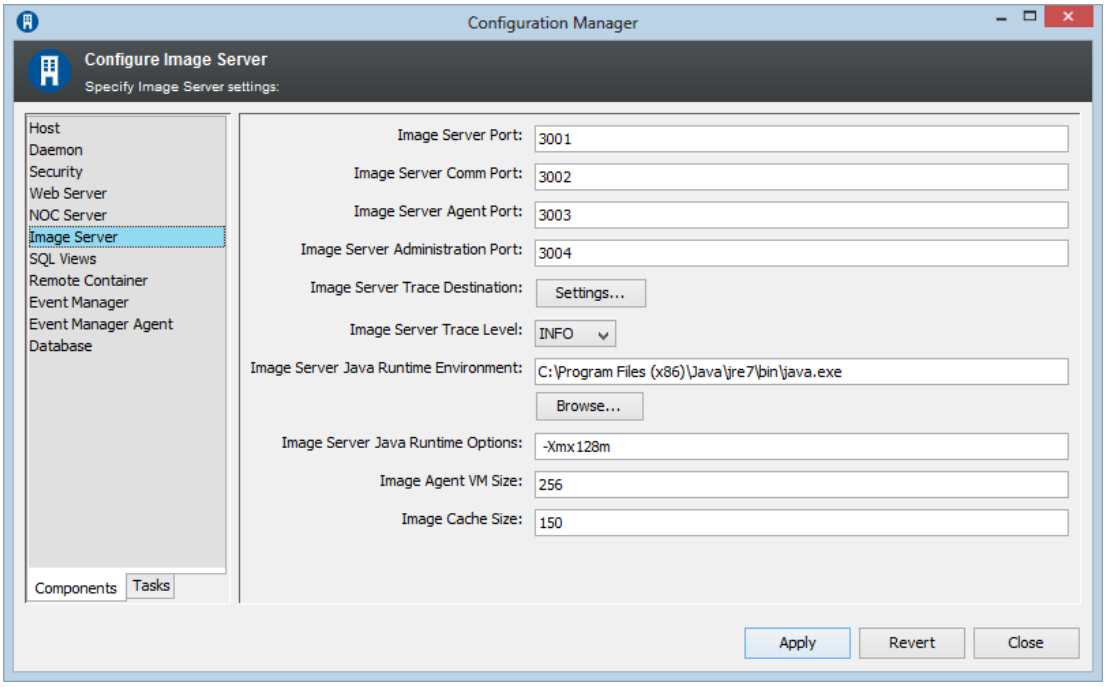


Table 2-6 *Configuration Manager Image Server Pane Settings*

Setting	Windows Default	UNIX Default	Description
Image Server Port	3001	3001	The port to handle external communications between a Web client and the Image server. See Chapter 4, “Configuring Network Communication Settings,” on page 49.

Setting	Windows Default	UNIX Default	Description
Image Server Comm Port	3002	3002	The port to handle internal communications between Operations Center and the Image server. See Chapter 4, “Configuring Network Communication Settings,” on page 49.
Image Server Agent Port	3003	3003	The port used by the Image server agent. See Chapter 4, “Configuring Network Communication Settings,” on page 49.
Image Server Administration Port	3004	3004	The port used to access the administration Web application for the Image server. See Chapter 4, “Configuring Network Communication Settings,” on page 49.
Image Server Trace Destination	N/A	N/A	The settings for the Image server trace logs. See Chapter 5, “Configuring Trace Logs,” on page 57.
Image Server Trace Level	INFO	INFO	Controls how much information is passed to the Image server trace logs. See Chapter 5, “Configuring Trace Logs,” on page 57.
Image Server Java Runtime Environment	`C:\OperationsCenter_install_path\jre\bin\java.exe`	`/OperationsCenter_install_path/jre/bin/java.exe`	Executable for the Java Runtime Environment for the Image server. Click *Browse* to navigate to the location of the JRE. See Chapter 6, “Configuring Java and Memory,” on page 63.
Image Server Java Runtime Options	`-server -Xmx128m`	`-server -Xmx128m`	Options for running the Image server. Usually does not need to be changed. See Chapter 6, “Configuring Java and Memory,” on page 63.
Image Agent VM Size	64	64	Specify VM size for the image agent.
Image Cache Size	150	150	Specify cache size for the image agent, in the number of images.

2.2.7 SQL Views Pane

The section describes Configuration Manager settings for SQL Views.

Figure 2-7 shows the SQL Views pane that contains configuration options for setting the SQL Views port. Table 2-7 provides information about configuring these options.

Figure 2-7 *Configuration Manager SQL Views Pane on the Components Tab*

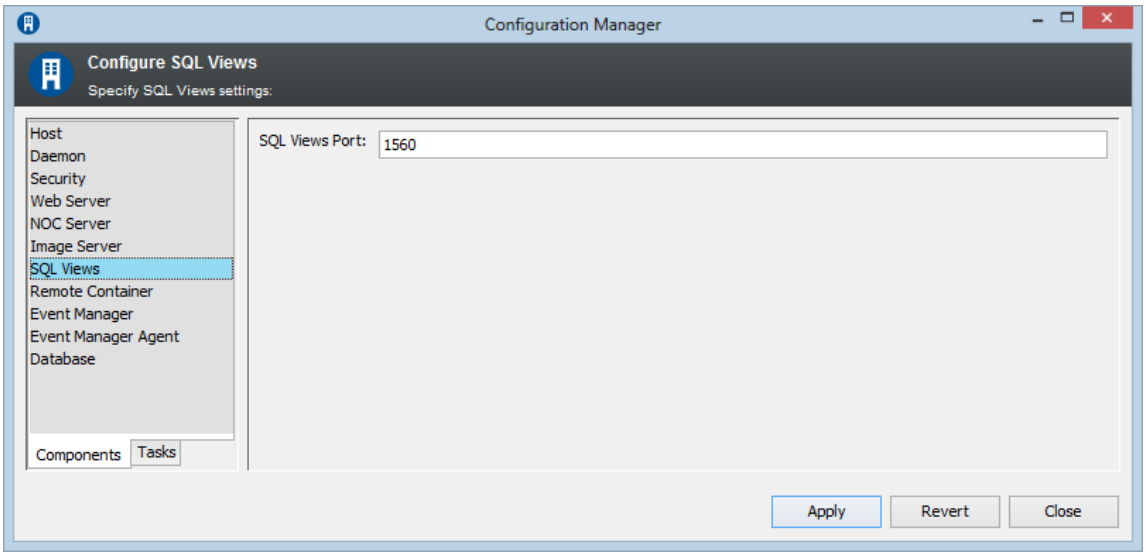


Table 2-7 *Configuration Manager SQL Views Pane Settings*

Setting	Windows Default	UNIX Default	Description
SQL Views Port	1560	1560	The port used by SQL Views for access to the Operations Center server. See Chapter 4, "Configuring Network Communication Settings," on page 49. For more information, see the *Operations Center 5.5 SQL Views Guide*.

2.2.8 Remote Container Pane

The Remote Container feature off-loads from the Operations Center server the resources required to configure and maintain multiple adapter instances.

Figure 2-8 on page 31 shows the Remote Container pane that contains configuration options for the remote container runtime environment and options, and log files. Table 2-8 on page 31 provides information about configuring these options.

For more information about the Remote Container, see the *Operations Center 5.5 Adapter and Integration Guide*.

Figure 2-8 *Configuration Manager Remote Container Pane on the Components Tab*

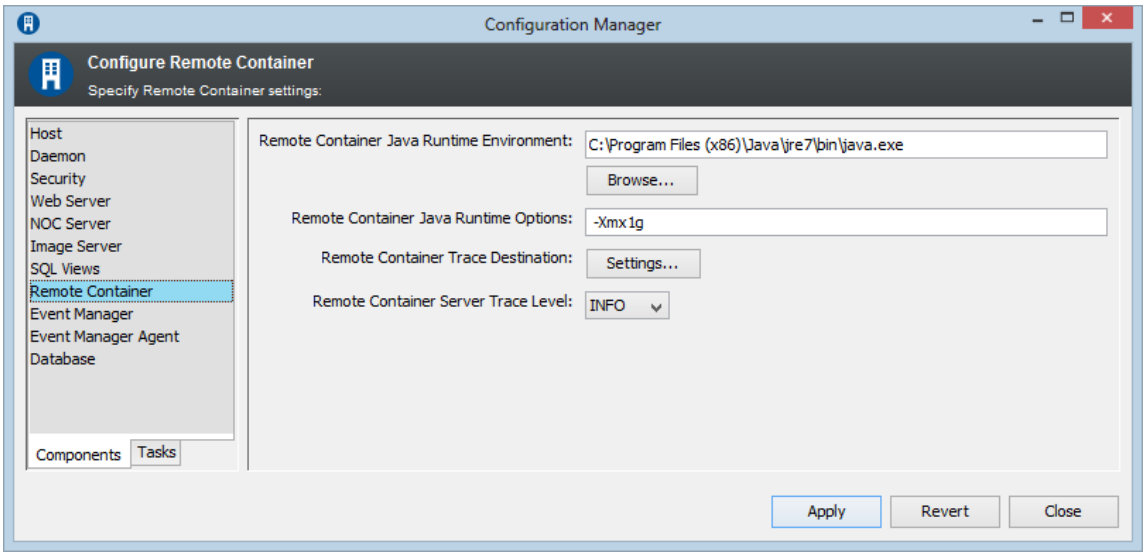


Table 2-8 *Configuration Manager Remote Containers Pane Settings*

Setting	Windows Default	UNIX Default	Description
Remote Container Java Runtime Environment	N/A	N/A	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. See Chapter 6, "Configuring Java and Memory," on page 63.
Remote Container Java Runtime Options	-server -Xmx512m	-server -Xmx512m	Options for running the Remote Container server. Usually does not need to be changed. See Chapter 6, "Configuring Java and Memory," on page 63.
Remote Container Trace Destination	N/A	N/A	The settings for the Remote Container server trace logs. See Chapter 5, "Configuring Trace Logs," on page 57.
Remote Container Server Trace Level	INFO	INFO	Controls how much information is passed to the Remote Container server trace logs. See Chapter 5, "Configuring Trace Logs," on page 57.

2.2.9 Event Manager Pane

The section describes Configuration Manager settings for Event Manager agents.

Figure 2-9 on page 32 shows the Event Manager pane that contains configuration options for the Event Manager runtime environment and options, and log files. Table 2-9 on page 32 provides information about configuring these options.

For more information about Event Manager, see the *Operations Center 5.5 Event Manager Guide*.

Figure 2-9 *Configuration Manager Event Manager Pane on the Components Tab*

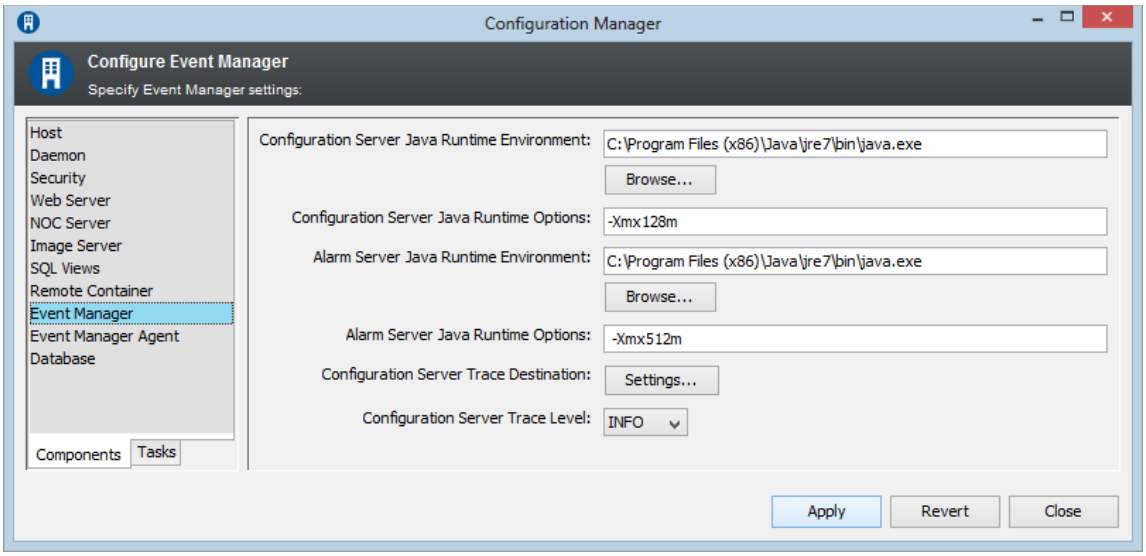


Table 2-9 *Configuration Manager Event Manager Pane Settings*

Setting	Windows Default	UNIX Default	Description
Configuration Server Java Runtime Environment	`C:\OperationsCenter_install_path\jre\bin\java.exe`	`/OperationsCenter_install_path/jre/bin/java.exe`	Executable for the Java Runtime Environment for the Event Manager Agent Manager. Click *Browse* to navigate to the location of the JRE. For more information about configuring Java and Memory, see Chapter 6, “Configuring Java and Memory,” on page 63.
Configuration Server Java Runtime Options	`-server -Xmx128m`	`-server -Xmx128m`	Options for running VM for the Event Manager Agent Manager server. Usually does not need to be changed. For more information about configuring Java and Memory, see Chapter 6, “Configuring Java and Memory,” on page 63.
Alarm Server Java Runtime Environment	`C:\OperationsCenter_install_path\jre\bin\java.exe`	`/OperationsCenter_install_path/jre/bin/java.exe`	Executable for the Java Runtime Environment for the Event Manager Alarm Server. Click *Browse* to navigate to the location of the JRE. For more information about configuring Java and Memory, see Chapter 6, “Configuring Java and Memory,” on page 63.
Alarm Server Java Runtime Options	`-server -Xmx256m`	`-server -Xmx256m`	A setting for VM for the Event Manager Alarm server. Usually does not need to be changed. For more information about configuring Java and Memory, see Chapter 6, “Configuring Java and Memory,” on page 63.

Setting	Windows Default	UNIX Default	Description
Configuration Server Trace Destination	*\OperationsCenter_install_path*\logs	*/OperationsCenter_install_path*/logs	The location in which to save the Event Manager server trace log files. For more information about configuring trace logs, see Chapter 5, “Configuring Trace Logs,” on page 57.
Configuration Server Trace Level	INFO	INFO	These settings control how much information is passed to the Event Manager server trace logs. For more information about configuring trace logs, see Chapter 5, “Configuring Trace Logs,” on page 57.

2.2.10 Event Manager Agent Pane

The section describes Configuration Manager settings for Event Manager agents.

Figure 2-10 on page 33 shows the Event Manager Agent pane that contains configuration options for the Event Manager Agent runtime environment and options, ports, and log files. Table 2-10 on page 34 provides information about configuring these options.

For more information about Event Manager and Event Manager agents, see the *Operations Center 5.5 Event Manager Guide*.

Figure 2-10 *Configuration Manager Event Manager Agent Pane on the Components Tab*

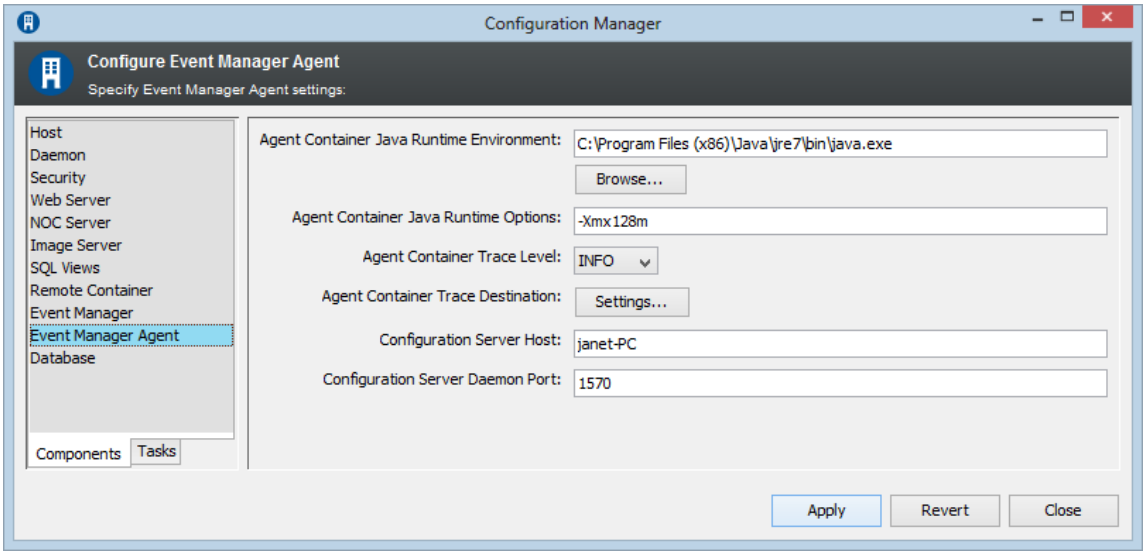


Table 2-10 *Configuration Manager Event Manager Agent Settings*

Setting	Windows Default	UNIX Default	Description
Agent Container Java Runtime Environment	`C:\OperationsCenter_install_path\jre\bin\java.exe`	`/OperationsCenter_install_path/jre/bin/java.exe`	Executable for the Java Runtime Environment for the Event Manager Agent Manager. Click *Browse* to navigate to the location of the JRE. For more information about configuring Java and Memory, Chapter 6, "Configuring Java and Memory," on page 63.
Agent Container Java Runtime Options	`-server -Xmx128m`	`-server -Xmx128m`	Options for running VM for the Event Manager Agent Manager server. Usually does not need changing. For more information about configuring Java and Memory, Chapter 6, "Configuring Java and Memory," on page 63.
Agent Container Trace Level	INFO	INFO	These settings control how much information is passed to the Event Manager agent trace logs. For more information about configuring trace logs, see Chapter 5, "Configuring Trace Logs," on page 57.
Agent Container Trace Destination	`\OperationsCenter_install_path\logs`	`/OperationsCenter_install_path/logs`	The location in which to save the Event Manager agent trace log files. For more information about configuring trace logs, see Chapter 5, "Configuring Trace Logs," on page 57.

Setting	Windows Default	UNIX Default	Description
Configuration Server Host	*IP address for localhost*	*IP address for localhost*	IP address for the Eve Configuration Manager for the agent. When an agent is installed with Operations Center or a Remote Container, this defaults to the local server. When agent is stand-alone, it must be populated.
Configuration Server Daemon Port	1570	1570	The daemon port on the server specified above.

2.2.11 Database Pane

The section describes Configuration Manager settings for the database.

Figure 2-11 shows the Database pane that contains configuration options for loading custom Java class files, and warehouse write mode. Table 2-11 provides information about configuring these options.

Figure 2-11 *Configuration Manager Database Pane on the Components Tab*

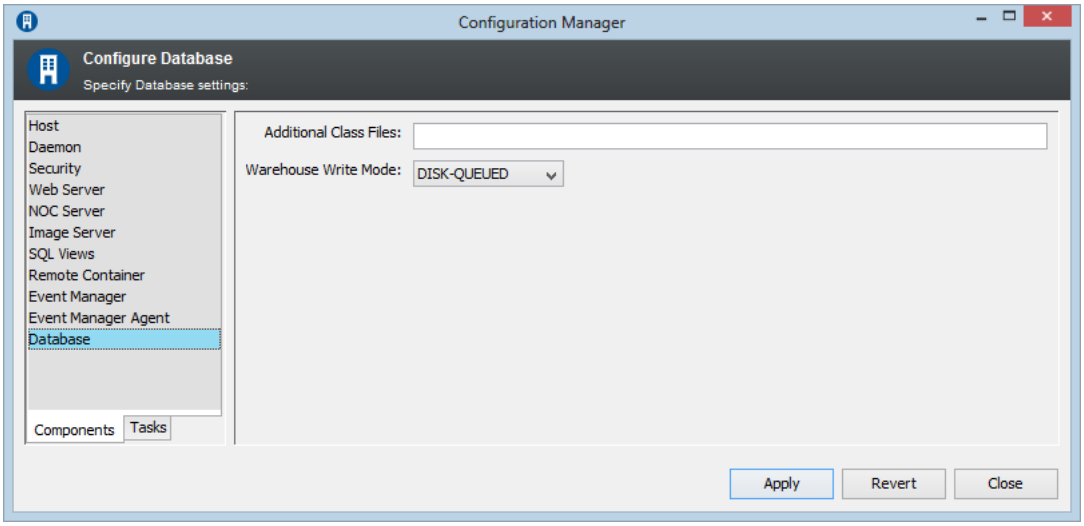


Table 2-11 *Configuration Manager Database Pane Settings*

Setting	Windows Default	UNIX Default	Description
Additional Class Files	N/A I	N/A	Comma-separated custom Java class files to load when the Operations Center server starts. These files should be placed in the `/OperationsCenter_install_path/classes/ext` directory. Loading class files is only necessary if you do custom development.
Warehouse Write Mode (Only Affects Clustering)	DISK-QUEUED	DISK-QUEUED	Select *MEMORY-QUEUED* for one Operations Center server in a clustered environment to update the database for the Service Warehouse. Select *DISK-QUEUED* to have an internal disk-based queued repository used for the BSW to queue data locally instead of using the backup repository. The backup repository has two main drawbacks: it uses more memory and it is nondeterministic when queueing to disk occurs. Disk-Queued can be used in either clustered mode or nonclustered mode. For clusters, it allows the primary warehouse writer to float between members of the cluster. See "Implementing a High Availability Solution" in the *Operations Center 5.5 Server Installation Guide*. If used in nonclustered mode, the write throughput speed for alarms takes about a 10% performance hit, and performance for nonalarm BSW writes increases. The benefit to using the Disk-Queued mode in nonclustered mode is improved fault-tolerance to outages of the server (data is always buffered locally on disk before sending to the BSW), and drastically reduced memory usage for the server (no in-memory queuing occurs).

2.3 Setting Up Configuration File Backups

Changes made in Configuration Manager are stored in properties files located in the `/OperationsCenter_install_path/config` directory. Within this folder is a `template` directory that contains templates of some of the properties files.

After finalizing changes in Configuration Manager, consider making backup copies of the configuration files in case a change is made in Configuration Manager that is later deemed unacceptable. Also consider making a backup copy of the template file that originally used to create the configuration file, since the template files might be updated when you install a new version or a service pack for Operations Center.

To persist configuration file changes:

1. Locate the */OperationsCenter_install_path*/config/template directory.
2. Make a backup copy of the original template version of the file, adding a new extension such as .original to designate it as the original copy of the file.
3. Make a backup copy of the changed configuration file, adding a new extension such as .modified to designate it as the modified version of the file.

   IMPORTANT: Do not copy either file from its folder to the other folder. They are not identical. Copying them can cause the server to fail to start.

4. Run Configuration Manager, then click the *Apply* button.
5. Check the final version of the file (the one the server actually uses, not the template version) to verify that the changes are included.
6. Continue with Section 2.4, "Making Custom Changes," on page 37.

2.4 Making Custom Changes

By default, the */OperationsCenter_install_path*/config/Formula.properties file stores standard configuration settings. Most of these configurations can be set and changed using the Configuration Manager.

However, some settings are unique to the Formula.properties file and cannot be configured using Configuration Manager. To add to or modify these customized settings, it is best to create and use a Formula.custom.properties file to ensure retention when the Formula.properties is overwritten during product reinstallations, upgrades, or when settings are changed using Configuration Manager.

Custom configuration settings saved in the Formula.custom.properties file supersede values in the Formula.properties file. In other words, if the same option is specified in both the Formula.custom.properties and Formula.properties files, Operations Center uses the value as specified in the Formula.custom.properties file.

To create a custom configuration file:

1. In a text editor, create a new file in the */OperationsCenter_install_path*/config directory and save it with the following name:

   Formula.custom.properties

   Verify that no additional file extension is added in the file name.
2. Add properties as necessary by copying and pasting property entries from the Formula.properties file and editing them as desired.
3. Stop and restart the Operations Center server for changes to take effect.

   For instructions on stopping the Operations Center server, see Section 3.4, "Manually Stopping Operations Center Components," on page 44.

   For instructions on starting the Operations Center server, see Section 3.3, "Manually Starting Operations Center," on page 44.

3 Configuring Operations Center Start Conditions

A daemon program called `mosdaemon` starts automatically as part of the Operations Center installation and controls the start of the Operations Center server as well as other Operations Center components. An auto-start utility can automatically start the Operations Center server and other components.

To configure Operations Center start conditions:

- Section 3.1, “Configuring mosdaemon,” on page 39
- Section 3.2, “Configuring an Auto-Restart,” on page 41
- Section 3.3, “Manually Starting Operations Center,” on page 44
- Section 3.4, “Manually Stopping Operations Center Components,” on page 44
- Section 3.5, “Configuring Web Server Start and Stop,” on page 45
- Section 3.6, “Manually Starting or Stopping the Image Server,” on page 47

3.1 Configuring mosdaemon

A daemon is a program that runs continuously and exists for the purpose of handling periodic service requests that a computer system expects to receive. The daemon program forwards the requests to other programs or processes as appropriate.

The Operations Center daemon service, known as `mosdaemon`, controls the following two features:

- Licensing service with which all components must register
- Launching and contacting service programs

  The Operations Center server is an example of a service program launched by the daemon.

To configure `mosdaemon`:

- Section 3.1.1, “Starting mosdaemon,” on page 39
- Section 3.1.2, “Configuring Access to mosdaemon,” on page 41
- Section 3.1.3, “Checking the mosdaemon Status,” on page 41

3.1.1 Starting mosdaemon

To start mosdaemon:

- “Starting mosdaemon on Windows” on page 40
- “Starting mosdaemon on UNIX” on page 40
- “Automating the mosdaemon Startup” on page 40

Starting mosdaemon on Windows

On Windows-based servers, `mosdaemon` starts automatically as a service program. Running `mosdaemon` as a Windows service is preferable because the `mosdaemon` process and all programs it launches terminate when the user logs off.

To manually start `mosdaemon`, do one of the following:

- Click *Start* > *Programs* > *NetIQ Operations Center* > *Start NetIQ Operations Center* on the desktop.

  Starting Operations Center via the Windows Start menu, starts the Operations Center server locally, but not as a service. Therefore, when you log out, the Operations Center server stops.
- From the */OperationsCenter_install_path*`/bin` directory, type `mosdaemon` at the command prompt.

Starting mosdaemon on UNIX

On UNIX-based servers, `mosdaemon` needs to be integrated into the operating system's startup scripts, if required.

Running `mosdaemon` on UNIX servers is platform specific. Refer to the operating system documentation for information about creating an initiation script.

Automating the mosdaemon Startup

The daemon can be configured to restart automatically in the event of a catastrophic failure.

To automate the `mosdaemon` setup:

1. In a text editor, open the */OperationsCenter_install_path*`/config/monitor.properties` file.

```
# For Unix,
# Monitor.DaemonRestartCommand=/bin/sh -c /opt/Formula/bin/mosdaemon
#
# For Windows service,
# Monitor.DaemonRestartCommand=net start "Managed Objects Daemon"
#
# For generic use,
# Monitor.DaemonRestartCommand=mosdaemon
```

2. Remove the pound sign (#) next to the appropriate UNIX, Windows, or generic `Monitor.DaemonRestartCommand`.

   This enables automatic restart of the Operations Center daemon on your chosen operating system.

3.1.2 Configuring Access to mosdaemon

By default, any host can connect to the Operations Center `mosdaemon`. It is possible to prevent any host machine from accessing the Operations Center `mosdaemon`. When the access restriction security option is enabled, access is denied to all hosts except for those hosts whose IP addresses are specified in the `mosdaemon.properties` file.

To restrict access to `mosdaemon`:

1. From a text editor, open the */OperationsCenter_install_path*`/config/mosdaemon.properties` file.
2. Enter `CORBA.Allow=`*`host_IP_address`* at the command prompt.

   Use the IP address for host machines allowed to access to the Operations Center `mosdaemon`.
3. Save the file.
4. Stop and restart the Operations Center server.

3.1.3 Checking the mosdaemon Status

Enter `mosstatus` at a command prompt. The operational status of the following programs started by the `mosdaemon` program is returned:

- Operations Center server (uptime, version, license expiry date, and patch level)
- Operations Center server database
- Database connections
- Service Warehouse
- Adapters
- Users currently logged in

3.2 Configuring an Auto-Restart

The Auto-Restart monitoring utility ensures that Operations Center software and its ancillary processing run continuously. The Auto-Restart utility probes Operations Center and other processes launched by the daemon process at periodic intervals. If it determines that a process can no longer be accessed using CORBA or that a process has stopped, it attempts to restart the process. Optionally, the Auto-Restart utility can be configured to restart the daemon process.

The Auto-Restart utility can be launched using one of the following methods, depending on the purpose for using it:

- As a Windows service
- Under daemon control
- Using the mosmonitor script under UNIX

A potential conflict exists for UNIX systems that run cron jobs that detect when Operations Center software is down and restart it. The cron jobs must be disabled before using the Operations Center Auto-Restart utility.

To configure an auto-start:

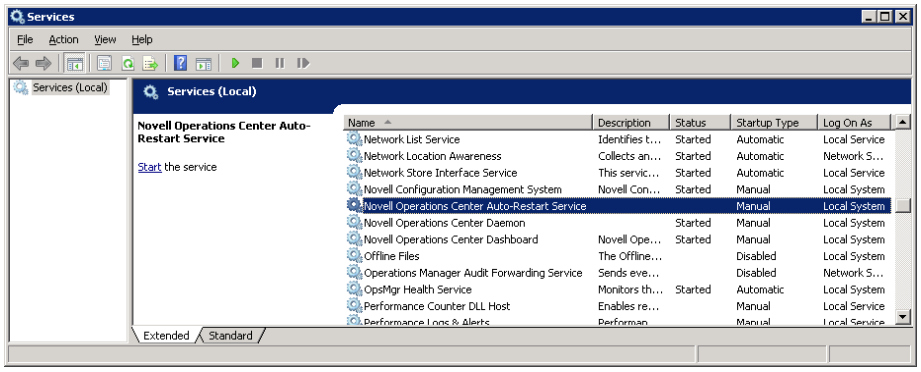
- Section 3.2.1, “Configuring the Windows Service,” on page 42
- Section 3.2.2, “Configuring Daemon Control,” on page 43
- Section 3.2.3, “Stopping mosmonitor from Automatically Restarting Operations Center,” on page 44

3.2.1 Configuring the Windows Service

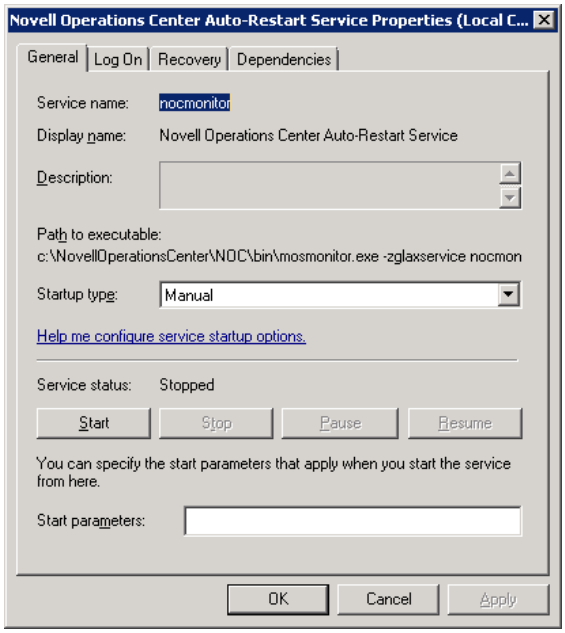
By default, the Auto-Restart utility is installed as a Windows service and is set to run automatically. Under Windows and UNIX, the daemon can be restarted if necessary because the Auto-Restart utility runs as an independent process.

To launch the Auto-Restart utility as a Windows Service:

1. From the desktop, click *Start* > *Control Panel*.
2. Double-click *Administration Tools*.
3. Double-click *Services* to display the Services window:



4. Double-click *NetIQ Operations Center Auto-Restart Service*.



5 From *Startup Type* drop-down list, select one of the following options:

Startup Type	Description
Automatic	The Auto-Restart service automatically starts when Windows starts. If running, the Auto-Restart service automatically starts Operations Center software when the server stops running.
Manual	The Auto-Restart service does not start when Windows starts, but can be manually started and stopped. Use a command line to start the Operations Center server.
Disable	The Auto-Restart service is disabled. To start the Operations Center server, reset this option to *Manual* and use a command line to start it.

6 Click the *Start* button.

7 Click *OK* to start the service.

3.2.2 Configuring Daemon Control

Launch the Auto-Restart utility under daemon control if network stability problems or environmental issues cause the Operations Center Java process to exit. Using this method, the daemon cannot be restarted because the Auto-Restart process is launched as a child process of the daemon. The Auto-Restart monitoring process inherits the daemon's resources if the daemon exists for any reason.

To configure the Auto-Restart utility to start under daemon control:

1 In Configuration Manager, select the *Daemon* pane on the *Components* tab.

2 In the *Automatically Start Servers* field, enter the server names in the list of servers that start automatically from the daemon.

By default, this is set to `Image Formula`.

Use a space to separate server names. Append with `EveConfigManager EveAgentManager EveAlarmServer` to start Event Manager components, or append with *`RemoteContainer_name`* to start Remote Containers on the same server.

For information on Event Manager, see *Operations Center 5.5 Event Manager Guide*. For information on Remote Containers, see "Using Remote Containers" in the *Operations Center 5.5 Adapter and Integration Guide*.

3 To pause the daemon between server startups, enter the keyword `pause(`*`n`*`)` between the server names, where *n* is the number of seconds.

If you enter 30 for *n*, `Daemon.trc` will contain the message: `Pausing for 30 seconds`.

Entering pause Formula without specifying any seconds pauses the daemon for one second.

4 Click *Save* to save all of the changes in Configuration Manager.

Regardless of whether the database setting is specified for this property, if there is an active database definition set up to use the Operations Center embedded database for Configuration Storage, the embedded database starts when the server starts.

3.2.3 Stopping mosmonitor from Automatically Restarting Operations Center

The `mosmonitor` service ensures that Operations Center software is running. If the Operations Center process stops, or if the daemon process stops, `mosmonitor` automatically restarts Operations Center software. However, you can shut down `mosmonitor` so that it won't automatically restart Operations Center.

To shut down the `mosmonitor` service, enter `mosmonitor -shutdown` at a command prompt.

The `daemon.trc` file logs that the monitor service has been shut down.

3.3 Manually Starting Operations Center

It is possible to start the Operations Center software manually. You can use a manual startup to avoid a potential problem for when you manually stop Operations Center software and do not want the Auto-Restart utility to immediately restart Operations Center software.

To manually start Operations Center:

1. Enter `mossstart` *`component`* at a command prompt, where *component* is the name of the component to start.

   For example, enter `mosstart` *`orb_server_name`* at a command prompt.

   Replace *orb* with the ORB name or with the Event Manager.

   Replace *server_name* with the hostname of the server where the ORB software is resident.

   If you only installed Operations Center software, no ORB software is running.
2. Because the format of the command used to start the Operations Center server depends on whether it is configured to start automatically, do one of the following:
   - If the Operations Center server is configured to start automatically, enter `mosdaemon` at the command prompt.
   - If the Operations Center server if it is configured to start manually, enter `mosdaemon` at a command prompt, press Enter, then enter `mosstart Formula`.

   The Operations Center software starts.
3. Before attempting to connect to Operations Center, determine whether Operations Center is running using the `mosstatus` command.

   The Operations Center server can take a few minutes to start.

3.4 Manually Stopping Operations Center Components

You can manually stop Operations Center, its individual components, including with a delay:

- Section 3.4.1, "Immediately Stopping All of Operations Center," on page 45
- Section 3.4.2, "Stopping Operations Center with a Delay," on page 45
- Section 3.4.3, "Stopping Only a Particular Component," on page 45

3.4.1 Immediately Stopping All of Operations Center

To stop `mosdaemon`, the Operations Center software, and the database servers:

Enter `mosstop -shutdown` at the command prompt.

All Operations Center software and database servers are stopped, which also stops all Operations Center adapters.

3.4.2 Stopping Operations Center with a Delay

There are two command line arguments to provide delays in shutting down Operations Center:

- To wait until all servers are shut down before returning control to the command prompt, enter `mosstop -shutdown -wait` at the command prompt.
- To wait a specific number of seconds for the servers to shut down before returning control to the command prompt, enter `mosstop -shutdown -wait -timeout` *`nn`* at the command prompt, where *nn* is the number of seconds to wait for each server to terminate.

3.4.3 Stopping Only a Particular Component

To stop a particular component, enter the `mosstop` *`component`* command, where *component* is the component name.

For example, enter `mosstop` *`orb server_name`* at the command prompt. Replace *orb* with the ORB name or with the Event Manager, and replace *server_name* with the hostname of the server where the ORB software resides.

If you only installed Operations Center software, no ORB software is running.

3.5 Configuring Web Server Start and Stop

The Operations Center Web server should only be started and stopped using the menu option in the Operations Center console.

IMPORTANT: You must stop and restart the Web server using only the commands in the console. Do not use any other method to shut down the Web server.

- Section 3.5.1, “Manually Starting or Stopping the Web Server,” on page 46
- Section 3.5.2, “Configuring Server Initialization and Messaging Settings,” on page 46

3.5.1 Manually Starting or Stopping the Web Server

To start or stop the Web server manually:

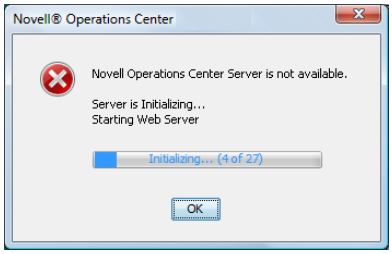
1. In the *Explorer* pane in the Operations Center console, expand *Administration* > *Server*.
2. Right-click *Web Server*, then select *Stop Web Server* or *Start Web Server*.

3.5.2 Configuring Server Initialization and Messaging Settings

Configuring server initialization and messaging settings includes:

- Starting the Web server before the Operations Center software initializes
- Logging status messages

The `Formula.custom.properties` file is used for starting the Web server. To configure Web server settings, update this file by adding the options documented in the following table:

Option	Description
Server.initialization.start.WebServerEarly	If True, the Web server starts before the Operations Center server completely initializes. Starting the Web server before the server completely initializes causes a status dialog box to open when Operations Center console's attempt to connect before the Operations Center server fully initializes. To customize the messages shown in the following status dialog box, see the Server.initialization.showDetails and Server.initialization.showElementLoadProgress rows in this table: 
Server.initialization.showDetails	If True, status messages contain more detail information. If False, no status messages display in the status dialog box.
Server.initialization.showElementLoadProgress	If True, a percent complete message displays during initial load of the persistent Operations Center elements. Configure the percent complete status message to display the number of elements complete (*n*) and the percentage complete (*m*). If False, the percent complete status is not shown.
Server.status.logStatusMessages	If True, status messages write to the log file as Info messages.

For more information about using the `Formula.custom.properties` file to customize configuration options, see Section 2.4, "Making Custom Changes," on page 37.

3.6 Manually Starting or Stopping the Image Server

The Image server is automatically started with the `mosdaemon` command. However, to manually start or stop the image server:

1. From a command line on the Image server's console:
   - To manually start the server, enter `mosstart`.
   - To manually stop the server, enter `mosstop`.
2. To manually start or stop the Image server using the Image Server Administration console from any workstation, enter http://*hostname*:3004/casapp/administrator in a Web browser, where *hostname* is the hostname of the Image server, then:
   - To restart the server, select *Restart*.
   - To start the server, select *Start*.
   - To stop the server, select *Stop*.

4 Configuring Network Communication Settings

Configuring Operations Center for networking between components involves the following:

- Server IP address and hostname
- Ports

For information on how to secure comunications, refer to the *Operations Center 5.5 Security Management Guide*.

For configuring IP Addtress and port settings:

- Section 4.1, “Configuring Server IP Addressing,” on page 49
- Section 4.2, “Configuring Ports,” on page 51

4.1 Configuring Server IP Addressing

The IP address and hostname for the Operations Center server are both initially detected during the installation of the Operations Center software. If you change the IP address or hostname for the machine on which the Operations Center server is installed, you also need to change the IP address or hostname in the Updating Server Settings Using the Configuration Manager (page 17).

Also, if you set any custom properties or have any scripts running that use the IP address or hostname, be sure to make the appropriate change in those files.

The following sections cover the following related topics:

- Section 4.1.1, “Restricting Access by IP Address,” on page 49
- Section 4.1.2, “Configuring NAT,” on page 50

4.1.1 Restricting Access by IP Address

Restrict access to Operations Center components by IP address, including the remote server’s access to the Operations Center server, as well as Operations Center console connections.

To restrict access to Operations Center components by IP Address:

1 Using a text editor, open the `/OperationsCenter_install_path/config/Formula.custom.properties` file.

For more information about creating or editing the `Formula.custom.properties` file, see Section 2.4, “Making Custom Changes,” on page 37.

2 Add or edit the `CORBA.Allow` command to include the necessary IP addresses.

   For example:

   ```
   CORBA.Allow=206.55.26.20,206.55.26.21,206.55.26.23
   ```

   Use commas (without extra spaces) to separate multiple IP addresses.

   The IP addresses must be complete and cannot use wildcards.

3 Save the `Formula.custom.properties` file.

4 Restart the Operations Center server.

5 Continue with Section 4.1.2, "Configuring NAT," on page 50.

4.1.2 Configuring NAT

If you need to allow for Network Address Translation (NAT) devices in Operations Center, refer to the following topics for instructions:

- "Understanding NAT Devices in Operations Center" on page 50
- "Allowing for NAT Devices" on page 50

Understanding NAT Devices in Operations Center

A commonly deployed networking tool is NAT. Although these devices do not meet a strict standard, they are sometimes considered firewalls because they allow machines with nonroutable, private IP addresses to connect to other networks.

If a NAT device exists anywhere in the network topology between the server and any clients, you must make two configuration changes to the server.

Operations Center consoles and other Operations Center components issue a standard request to a target server when they want to communicate with it. This request travels over either the unsecured or secured Web port, and takes the form of a special URL. The originating Operations Center component parses the data returned by this request to determine which port number to use for further communication with the target. This same mechanism used to return port values also specifies the IP address of the server.

However, in the presence of a NAT device, the IP address published by the server is the IP address of the server from the private side of the NAT device, not the public side, and this IP address does not allow the client to successfully pass from the user's desktop through the NAT device to reach the server.

If the server has multiple network interface cards, the "host" settings usually list each network card interface in a value in the `Formula.properties` file named `CORBA.alternateNetworkInterfaces`. The setting for these alternate interfaces conflicts with the hostname setting because server references must be published with names instead of addresses. However, it is possible to specify a mixture of hostnames and IP addresses for `CORBA.alternateNetworkInterfaces` that allows more flexibility in resolving the server's location.

Allowing for NAT Devices

To configure for NAT devices:

1 Update the */OperationsCenter_install_path*`/config/Formula.custom.properties` file for the following properties and values:

Property Name	Set to	Explanation
ooc.iiop.numeric	`false`	If `true`, specifies a Nonsecure (Standard) HTTP server identifier as a numeric value. If `false`, uses a hostname instead of IP address.
ooc.fssl.numeric	`false`	If `true`, specifies a Secure HTTPS server identifier as a numeric value. If `false`, uses a hostname instead of IP address.

Although each of these properties is associated with a different server communication protocol, it is best to change both to `False` in order to be consistent.

For more information about using the `Formula.custom.properties` file to customize configuration options, see Section 2.4, "Making Custom Changes," on page 37.

2 Add the following two properties:

Property Name	Set to	Explanation
ooc.iiop.host	*the server hostname*	Specifies a Nonsecure (Standard) HTTP name to be used by the server.
ooc.fssl.host	*the server hostname*	Specifies a Secure HTTPS name to be used by the server.

After the next successful startup, the server begins publishing the contact information using the hostname specified above, instead of an IP address.

3 Verify that the changes are in effect for every individual desktop where a Operations Center console resides.

The method of DNS resolution in effect successfully resolves the server's hostname to the correct public address needed to pass from the user's desktop through the NAT device to reach the server.

4.2 Configuring Ports

The Operations Center server cannot know in advance the location of other Operations Center component such as the Operations Center console. For this reason, Operations Center consoles and other Operations Center components issue a standard request to a target server when they want to communicate with it. This request travels over either the unsecured or secured Web port, and takes the form of a special URL. The originating Operations Center component parses the data returned by this request to determine which port number to use for further communication with the target.

If the Operations Center console contacts the target server using secure HTTP, the port value returned is the bidirectional IIOP port for secure data. If the Operations Center console contacts the target server using unsecured HTTP, then the port value returned is the bidirectional IIOP port for standard HTTP.

The server bidirectional port number is stored in the `/OperationsCenter_install_path/html/classes/CORBA.properties` file.

The InterConnection adapter (ICA) connection mechanism is similar to that of the Operations Center client to server. When specifying adapter properties, specify the hostname and HTTP port of the other Operations Center server to which you are connecting. It is possible to use the ICA over the

HTTPS port. The ICA follows the same 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 describe how to configure ports:

- Section 4.2.1, "Identifying Ports in Use," on page 52
- Section 4.2.2, "Configuring a Operations Center Server Port Range," on page 52
- Section 4.2.3, "Configuring the Operations Center Port Assignments," on page 54

4.2.1 Identifying Ports in Use

In Table 4-1, an "X" in the first 3 columns identifies the different settings that are possible for *Client/Server Communications Mode*:

Table 4-1 *Ports Defined in the Server Properties*

Unsecured	Secured	Both	Name	Explanation
X	X	X	ooc.iiop.port	Unidirectional IIOP port
X	X	X	ooc.iiop.acceptor.iiop.port	Unidirectional IIOP port
	X	X	ooc.fssl.acceptor.RootPOAManager.port	Bidirectional IIOP for SSL
X			ooc.iiop.acceptor.RootPOAManager.port	Bidirectional IIOP for standard HTTP
		X	ooc.iiop.acceptor.bidirManager.port	Bidirectional IIOP for standard HTTP

These values only display in the Operations Center console; they are not reported in `formula.trc`.

To identify the ports currently in use on the server and the specific usage of each port:

1. In the *Explorer* pane in the Operations Center console, expand the root *Administration* element.
2. Right-click *Server*, then select *Properties*.
3. Click the *System Information* tab.
4. Page down to view the port values.
5. To change the selected ports and their associated names, do one of the following:
   - Select a new value for the *Client/Server Communication Mode*.
   - Modify the port range in */OperationsCenter_install_path*`/config/mosdaemon.properties`.

4.2.2 Configuring a Operations Center Server Port Range

Review the following to determine whether you need to change server port ranges:

- "Understanding Server Port Ranges" on page 53
- "Changing the Server Configuration for a Fixed External Port" on page 53

Understanding Server Port Ranges

The Operations Center server uses a range of consecutive ports for its internal and external communications. The exact number depends upon the number of Java Virtual Machine (JVM) processes configured to start.

The default values for the range are specified in */OperationsCenter_install_path*/config/mosdaemon.properties under the headings of:

Table 4-2 *Default Server Port Specifications*

Name	Value
MOSDaemon.ServerPortStart	2000
MOSDaemon.ServerPortEnd	3000

During every initialization, the server starts at the lower number specified in `ServerPortStart` and looks for a consecutive range of unused port equal to the number it needs. The selected values in this range remain fixed as long as the server remains active.

The default values described above are fine for everyday usage in most environments. However, the potential exists for the server to use a different range of ports each time the server starts. If the configuration of the physical server changes to allow another application to start before the server, the other application might acquire one or more of the ports at the beginning of the range that Operations Center software tries to use, thus forcing Operations Center software to use a different set of values. In situations with firewalls where the port values must be fixed and known, this is a problem.

It is possible to open specific ports. However, do not change the default values before carefully studying the current values and considering the potential effects of changes. Also note that making changes means accepting the responsibility of ensuring that the changes remain in effect through the application of patches, upgrades, and general maintenance of the server.

IMPORTANT: Do not change the default server port specification without carefully considering the potential impact.

Changing the Server Configuration for a Fixed External Port

If there is a requirement for the server's external port usage to remain fixed, such as the existence of closed and secure firewalls, it is necessary to change the server configuration:

1. On the server, open the */OperationsCenter_install_path*/config/daemon.ini file for editing.
2. Replace the `Port=`*nnnn* line, where *nnnn* is the numeric value, for the first port in the consecutive range of ports that Operations Center software should acquire.

   On the server, this port is the `ooc.iiop.port` and `ooc.iiop.acceptor.iiop.port`.

   To start correctly, the server must find the correct number of consecutive, available ports beginning with the specified port number.
3. Save the file.
4. Continue with Section 4.2.3, "Configuring the Operations Center Port Assignments," on page 54.

4.2.3 Configuring the Operations Center Port Assignments

Specific port assignments are made for some of the Operations Center components. Review the following sections for information on making the port assignments:

- "Web Server" on page 54
- "Experience Manager" on page 54
- "Image Server" on page 55
- "The Dashboard using RMI" on page 55
- "Web Services" on page 55
- "SQL Views" on page 56
- "Event Manager" on page 56
- "ORBs" on page 56

Web Server

The Web server ports on the Operations Center server allow access to the Operations Center server from the integrated Web server used for the Web servers used for access from the Operations Center console and dashboard. There are three possible ports to set on the Operations Center server:

- **Web server port (HTTP):** Unsecured TCP/IP port number. The default is 80 for Windows and 8080 for UNIX.
- **Web server port (HTTPS):** Secured TCP/IP port number. The default is 443 for both Windows and UNIX.
- **Web server shutdown port:** Opens a socket connection to listen to shutdown commands for the Web server. The default is 8005 for both Windows and UNIX.

The settings for these ports are set in Configuration Manager on the *Networking* pane under the *Tasks* tab. For instructions, see Updating Server Settings Using the Configuration Manager (page 17).

For more information about Web server ports and communications security, see "Communications Security" in the *Operations Center 5.5 Security Management Guide*

Experience Manager

The Experience Manager also uses a Web server port. The default is 8080, which might conflict with the default Web server port for the Manager Objects server for UNIX.

This is the only port definition that might conflict with Operations Center software.

You should use different ports for the individual Web servers.

For more information on Experience Manager, see the *Operations Center 5.5 Experience Manager Guide*.

Image Server

An Image Server is required by Operations Center to allow Web clients (including the Operations Center console and dashboard) to render dynamic and 3-D charts. There are four ports to configure for the Image server:

- **Server port:** The port to handle external communications between a Web client and the Image server. Because of firewall restrictions, the Web server's name is actually used to access the Image server and then all Image server traffic is then redirected to the Image server. The default is 3001 for both Windows and UNIX.
- **Communications port:** The port to handle internal communications between the Operations Center server and Image server. The default is 3002 for both Windows and UNIX.
- **Agent port:** The port used by the Image server agent.
- **Admin port:** The port used to access the administration application for the Web server.

The settings for these ports are set in Configuration Manager on the *Networking* pane under the *Tasks* tab. For instructions, see Updating Server Settings Using the Configuration Manager (page 17).

For more information about the Image Server port and communications security, see "Communications Security" in the *Operations Center 5.5 Security Management Guide*

The Dashboard using RMI

In order for the dashboard to communicate with Operations Center, there need to be two ports opened. The first is for the Remote Invocation Method (RMI) directory which is the contact point for the dashboard to communicate with Operations Center. The second is randomly assigned by this contact point for the actual communications to occur. Because the second port is not set to a specific port, it makes it nearly impossible to use a firewall between the dashboard and Operations Center.

The Remote Services (RMI) port is configured in the Operations Center Configurations Manager and defaults to 1099 for both Windows and UNIX. The second port for actual communications can be set adding the following line to the `config/Formula.custom.properties` file:

`mymo.rmi.port=`*`port_number_1`*`:`*`port_number_2`*

where, *port_number_1* is the first available port in a range of ports ending with *port_number_2*.

For more information about using the `Formula.custom.properties` file to customize configuration options, see Section 2.4, "Making Custom Changes," on page 37.

For more information about the dashboard, see the *Operations Center 5.5 Dashboard Guide*.

For more information about the RMI ports and communications security, see "Communications Security" in the *Operations Center 5.5 Security Management Guide*

Web Services

Third-party applications can use the Operations Center Web Services Application Programmer Interface (WASPI) for access to the Operations Center server. The Web services port must be configured and is 8084 by default for both Windows and UNIX.

For more information, see the *Operations Center 5.5 Web Services Guide*.

SQL Views

SQL Views provides functionality in Operations Center that allows for third-party applications to have read access to Operations Center data. To allow access to the Operations Center server, you must designate a SQL Views port. The default is 1560 for both Windows and UNIX. The port is set in the Updating Server Settings Using the Configuration Manager (page 17) on the *Networking* pane under the *Tasks* tab.

For more information, see the *Operations Center 5.5 SQL Views Guide*.

Event Manager

For the Event Manager product, create sources that are access points for the Event Manager agents to receive data from hosts. As part of the source creation, specify a port for the following types of sources: server sockets and client sockets. Make sure that these port selections do not conflict with the ports already in use for Operations Center software.

For more information about the Event Manager, see the *Operations Center 5.5 Event Manager Guide*.

ORBs

Operations Center software uses Object Request Brokers (ORBs) to facilitate communication between the adapters used with the Operations Center server and some third-party management systems. The Operations Center server uses a different mechanism to initiate communication with ORBs, taking advantage of the fact that ORBs wait for the server to initiate contact. General ORB behavior requires two open ports:

- Command-and-control from the server to the ORB
- Unidirectional IIOP from the ORB to the server

Table 4-3 lists the default ports used by the ORBs:

Table 4-3 *Default ORB Ports*

ORB	Default Ports
MCORB	1099
NvORB	1572
OvORB	1572
OVOORB	1578
TecORB	1576 and 1577
UniORB	1580

If the defaults are already in use, it is necessary to change the port numbers.

You must specify the command-and-control port in the adapter properties. This is the port where the ORB listens for contact from the server. It includes the port number for sending information to the server. All ORBs share the single unidirectional IIOP port defined on the server.

5 Configuring Trace Logs

Trace logs are available for the following Operations Center components:

- Daemon for the Operations Center server (mosdaemon)
- Operations Center server
- Event Manager server
- Image server
- Remote Container server

The trace log settings for the Operations Center daemon (mosdaemon) and the Operations Center server must be configured. The Event Manager server settings also must be configured if you use Event Manager. For more information, see the *Operations Center 5.5 Event Manager Guide*.

There are two options for the trace log settings:

- The trace logs for Operations Center components are set so the settings persist for each session of the Operations Center server and are applicable to Operations Center components
- Define logging settings so they are applicable only for the current session of the Operations Center server for a specific element

Additional auditing settings are also available on the Operations Center server. For more information, see the *Operations Center 5.5 Security Management Guide*.

The following sections describe the options available for trace log files:

- Section 5.1, "Understanding Trace Logs and Content," on page 57
- Section 5.2, "Changing the Persistent Component Log Settings," on page 58
- Section 5.3, "Configuring Current Session Element Log Settings," on page 61

5.1 Understanding Trace Logs and Content

The trace logs store error messages according to specified parameters and are located in the `/OperationsCenter_install_path/logs` directory. Table 5-1 describes the trace log files that are configured during the Operations Center installation process:

Table 5-1 *Operations Center Trace Logs*

Trace Log	Description
Daemon Trace Log (`daemon.trc`)	Tracks all activity of the Operations Center daemon service, such as the daemon starting time and its stopping time.
Operations Center server Trace Log (`formula.trc`)	Tracks all activity of the server such as when the server starts, which components are running on the server, when the server stops, and so on.

Trace Log	Description
Operations Center Event Manager Trace (`eve.trc`)	Tracks all Event Manager activities such as whether all components of the Event Manager are running, stopped, and so on.
Database Trace Log (`database.trc`)	Tracks all database activities for the Operations Center databases. Track any error messages generated by the Operations Center database server and the mosdaemon service. If Operations Center is configured to use an external database and not the Operations Center embedded database (the default) for the Configuration Storage database, the following message displays in the `database.trc` log: `No integrated database servers defined. Event will be persisted to defined external databases.` This message is not an error.
Image Server Trace Log (`image.trc`)	Tracks Image server activities. For more information, see Section 9.5, "Using the Image Server," on page 137.
Remote Container Server Trace Log (`$MOSServerName.trc`)	The name of the trace log is the name of the Remote Container server.
Operations Center Service Level Manager (SLM) Log (`hyper.trc`)	Tracks SLM engine activities. Track any errors generated by the SLM query engine during priming, real-time monitoring, and reporting activities. For more information on SLM, see the *Operations Center 5.5 Service Level Agreement Guide*.

Use Notepad or any text editor to view them. For example, if there are problems starting the Operations Center software, view the Operations Center server trace file (`/`*`OperationsCenter_install_path`*`/logsformula.trc`).

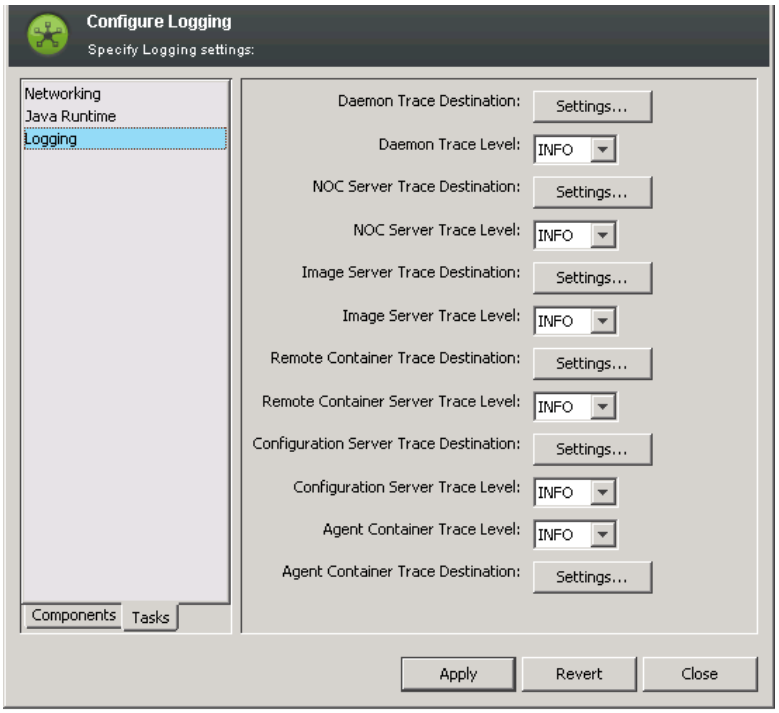
5.2 Changing the Persistent Component Log Settings

The trace logs can be set so the settings persist for each session of the Operations Center server and are applicable to Operations Center components. The log levels and other log settings can be set directly in the `Formula.custom.properties` file by setting the `log4j.category` (such as `log4j.category.Server.DirectoryService`).

For information about using the `Formula.custom.properties` file to customize configuration options, see Section 2.4, "Making Custom Changes," on page 37.

The trace logs are also configurable in the Operations Center Configuration Manager by setting the trace destination and the trace level from the *Logging* pane under the *Tasks* tab. For more information about the Configuration Manager, see Chapter 2, "Updating Server Settings Using the Configuration Manager," on page 17.

Figure 5-1 *Configuration Manager Logging Pane under Tasks Tab*



These same settings are also available under the *Components* tab on a pane that is for the specific component. For example, the *Server Trace Destination* and *Server Trace Level* settings also appear on the *Server* pane under the *Components* tab. For more information about specifying the trace destination and trace level settings:

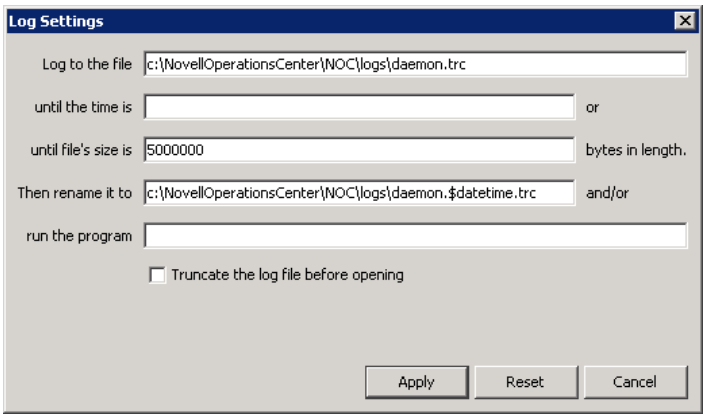
- Section 5.2.1, “Trace Destination,” on page 59
- Section 5.2.2, “Trace Level,” on page 61

5.2.1 Trace Destination

The trace destination sets both the location and name of the log file and other log file settings.

To access these settings, click *Settings* for the appropriate option on the *Logging* pane of the *Tasks* tab in Configuration Manager.

Figure 5-2 *Trace Destination Log Settings for the Operations Center Daemon*



The following table details the configurable settings for trace logs.

Table 5-2 *Trace Log Settings*

Setting	Default	Description
Log to the file	`/`*`OperationsCenter_install_path`*`/logs/componentname.trc`	Current trace information is collected and stored in this file. Verify that the default reflects the current path to your `/`*`OperationsCenter_install_path`*`/logs` directory.
Until the time is		The cut-off time for collecting trace log data. Enter time such as 11:00:00 am (or format as specified by your internationalization settings).
Until the file's size is	5000000	The maximum file size for collecting trace log data.
Then rename it to	`/`*`OperationsCenter_install_path`*`/logs/`*`componentname`*`.$datetime.trc`	When the trace log file reaches the time cut-off or maximum file size, the trace file is renamed to this entry, and a new log file is started.
Run the program	`/`*`OperationsCenter_install_path`*`/logs/`*`componentname`*`.trc`	Any program can be run here, but the most common is a file compression application.
Truncate the log file before opening	Not checked	Check to replace the contents of the log file with new log messages. Clear the check box to append new log messages to the log file.

Trace log information is collected until the time given, or until the file size is reached — whichever comes first. If either the file size or time is left blank, new log files are created only when the remaining limit is reached.

5.2.2 Trace Level

The trace level setting controls how much information is passed to the trace logs as follows:

- **ERROR:** Logs error messages only
- **WARN:** Logs error and warning messages
- **INFO:** Logs informational, error, and warning messages
- **DEBUG:** Logs all information and as the most detailed setting should be used only to track down problems

`INFO` is the default for the trace level setting for all logs and is recommended when Operations Center is first installed.

5.3 Configuring Current Session Element Log Settings

Define logging settings so they are applicable only for the current session of the Operations Center server for a specific element. When experiencing problems with the server, set the debugging levels for server elements while the server is running. Logging settings are modified while the server is running and are reset to the persistent settings when the server is restarted.

These log settings are set in the Operations Center console, located in the *NetIQ Operations Center* > *Enterprise* > *Administration* > *Server* element. Right-click *Server*, then select *Logging*.

The trace level options are:

- *Fatal*
- *Debug*
- *Error*
- *Info*
- *Warn*

These levels of debugging can be applied to any element as the logging category.

There are two methods to setting the trace level and logging category:

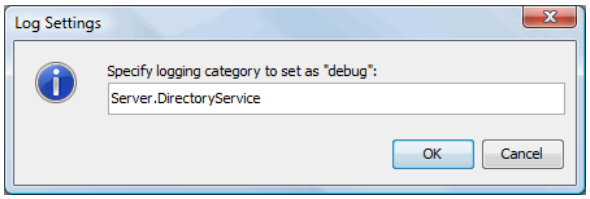
- Section 5.3.1, “Selecting the Trace Level, then the Server Element,” on page 61
- Section 5.3.2, “Selecting the Server Element, then the Trace Level,” on page 62

5.3.1 Selecting the Trace Level, then the Server Element

To set the trace level and logging category by setting the trace level first:

1. In the *Explorer* pane, expand *Administration*.
2. Right-click *Server*, then select *Logging*.
3. Click the desired *Save As* logging mode.

4 Specify the server element, then click *OK*.



5 Click *OK*.

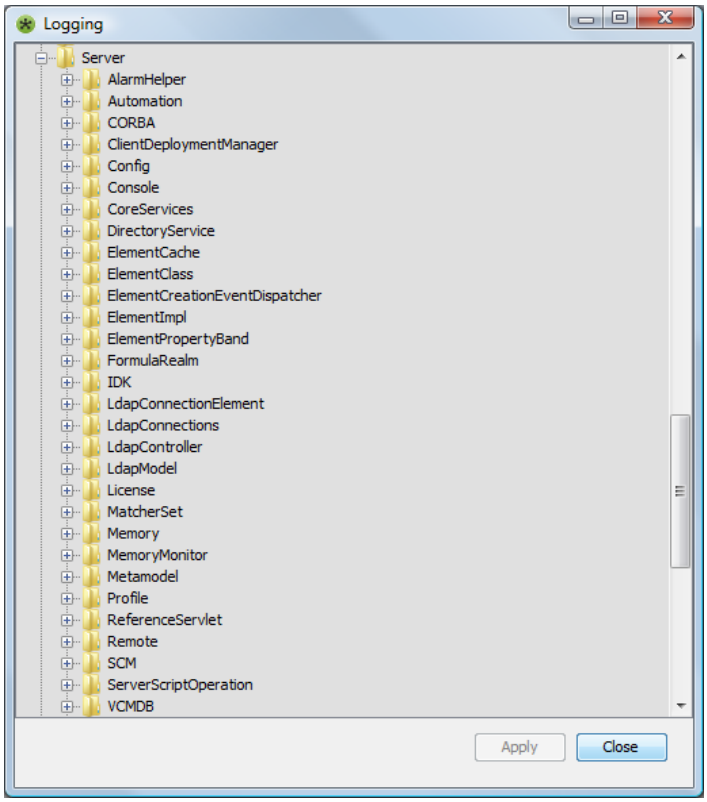
5.3.2 Selecting the Server Element, then the Trace Level

To set the trace level from the Server element:

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

2 Right-click *Server*, then select *Logging*.

3 Click *Display Categories*.



4 Expand *Logging Categories*, then explore and expand the logging items until you locate the desired logging category.

5 Right-click the desired logging item, then select the debugging level.

6 Click *Apply* to save the log setting, then *Close*.

Only after a change to any of the debugging levels occurs, does *Apply* become available.

6 Configuring Java and Memory

The Operations Center server and Remote Container server run in a Java Virtual Machine (JVM) that has memory allocation requirements. The Remote Container server runs in its own JVM. Operations Center and some of its components require a Java Runtime Environment (JRE) that must be configured.

- Section 6.1, “Configuring the Java Virtual Machine,” on page 63
- Section 6.2, “Configuring the Java Runtime Environment,” on page 68

6.1 Configuring the Java Virtual Machine

When installing a Operations Center server (or Remote Container server), the VM is configured with parameters for memory allocation. Operations Center installs with default values. These values can be changed in the Updating Server Settings Using the Configuration Manager (page 17). However, before adjusting the memory, it is important to understand how memory is allocated.

- Section 6.1.1, “About Java and Memory,” on page 63
- Section 6.1.2, “Understanding Windows Memory Restrictions,” on page 64
- Section 6.1.3, “Understanding Memory Allocation,” on page 64
- Section 6.1.4, “Understanding Memory Parameters,” on page 65
- Section 6.1.5, “Configuring Operations Center’s Use of Memory,” on page 66
- Section 6.1.6, “Resolving Out of Memory Errors in Trace Logs,” on page 67

6.1.1 About Java and Memory

In general, a program written in Java usually requires more memory than the same program written in C. However, programs written in Java can be more robust at an earlier phase of the development cycle, because of the design of the language and the runtime environment.

When running a program written in C, you do not specify how much memory that program uses. Most of these programs obtain additional memory from the operating system as they need it. There is no fixed upper limit, other than the operating system’s maximum allowance.

Java, in contrast, requires declaring the maximum amount of memory to use. This can be set to a very high number, such as 2 or 3 gigabytes, but there are reasons why this might not be a good idea.

When the Java virtual machine starts, it allocates a portion of memory from the operating system. The size of memory allocated is usually 8MB, unless it is told to allocate more.

The Java virtual machine has an inherent runtime overhead, which can amount to quite a bit of memory. There is no direct way to control the size of this overhead. Recognize that if a parameter is passed to the Java virtual machine instructing it to use 2 gigabytes of memory, the result can be, according to various utilities, that the virtual machine uses more than 2 GB. This is normal and should be expected.

If a maximum memory usage has been set, the virtual machine usually does not need all the memory right away and won't allocate it from the operating system. Instead, the virtual machine determines how much of its heap space allocated from the operating system should actually be used for Java objects.

If this is less than 30%, memory is released back to the operating system. If it is more than 70%, the virtual machine asks the operating system to give it another chunk of memory (usually a fairly large chunk). If the virtual machine has reached the maximum heap size provided, it does not ask the operating system for more memory.

6.1.2 Understanding Windows Memory Restrictions

Windows restricts application address space to two (2) gigabytes regardless of the amount of memory available. Java, which consists of heap and nonheap memory, requires a maximum size of 1.7 gigabytes that is a contiguous block. Often this amount of contiguous memory is not available. This causes the actual size to be between 1.2 and 1.5 gigabytes.

If you require more memory, you must use a 64-bit OS and JVM or use Linux on x86 to have 3 gigabytes heap memory available since these systems are not limited by Windows restriction.

6.1.3 Understanding Memory Allocation

Garbage collection is usually triggered by a Java program allocating an object as follows:

- **New generation (NG):** Where newly created objects are allocated; mostly short-lived objects.
- **Old generation (OG):** Where long-lived objects go.

Object allocation is actually fairly complicated and breaks down something like this:

- Java program makes request for N bytes.
- Object is allocated from the new generation, if space is available.
- If space is not available, a new generation collection is initiated.
- The object is allocated from the new generation if space is available.
- If space is not available, objects are transferred, by age, into the old generation according to heuristics.
- If no space is available in the old generation for the objects, a full garbage collection cycle is run. This compacts the old generation.
- If no space is available and the VM has not yet requested its maximum allocation from the operating system, it asks the operating system for more memory.

This is a simplified model of what happens, but it is fairly close to reality.

The Java VM maintains the NG and OG areas of the heap. The maximum amount of memory that the VM uses in the new generation can be controlled with the `-XX:MaxNewSize=`*`size`* parameter. This permits the VM to use extra memory for the new generation.

6.1.4 Understanding Memory Parameters

The following table lists the basic parameters used to control the behavior of the VM.

Table 6-1 *VM Memory Parameters*

Parameter	Description
`-Xmx`*`size`*	Sets the maximum amount of heap memory Java can use. For example, `-Xmx1536m`.
`-Xms`*`size`*	Sets the initial amount of heap memory Java allocates. For example, `-Xms1536m`.
`-Xmn`*`size`*	Sets the maximum size of the new generation.
`-XX:MaxPermSize=`*`size`*	Sets the maximum size of the permanent generation. For example, `-XX:MaxPermSize=100m`.

The `-Xms` parameter can instruct the virtual machine on how much memory to initially allocate. The `-Xmx` parameter instructs the virtual machine how much memory, at most, it is allowed to use.

In most Operations Center installations, use the `-Xmx` and `-XX:MaxNewSize` parameters, and possibly the `-Xms` parameter.

The following table lists the Windows and UNIX defaults for the memory parameters for the Operations Center components.

Table 6-2 *-Xmx<size> for Operations Center Components*

Components	Windows	UNIX
Daemon	128	256
Operations Center server	512	128
Operations Center client	256	512
Event Manager Configuration Manager*	128	256
Event Manager Alarm Server*	256	128
Event Manager Agent Manager*	128	256
Database server	128	128
Image server	128	128
Configuration Explorer	1	1
Remote Container server**	512	128

*For more information about the Event Manager, see the *Operations Center 5.5 Event Manager Guide*.

**The Remote Container server parameter is applicable only for an installation that is running as a Remote Container. For more information on Remote Container, see the *Operations Center 5.5 Adapter and Integration Guide*.

The default can be changed in the Updating Server Settings Using the Configuration Manager (page 17). It is the Java Runtime Option for each component and is listed on the *Java Runtime* pane under the *Tasks* tab.

6.1.5 Configuring Operations Center's Use of Memory

Before changing the memory parameters from the default values, determine how the Operations Center software uses memory. Monitor the `fsgc.log` file to determine what is being used. This file contains all garbage collection action.

Check the amount of physical memory on the machine — do not take logical or paged memory into consideration as these do not help Java. Find out the amount of memory that a Operations Center installation needs to run, then set the maximum memory size to a value larger than this. Then, select an appropriate new generation size.

Also check the `daemon.trc` file. If there is an OutOfMemoryError in the `daemon.trc` file originating from Operations Center software, it is usually an indicator that too little memory is provided to the Java VM. If Operations Center software tries to allocate a very large amount of memory by mistake or there are certain kinds of IO errors, you also receive an OutOfMemory message.

To determine how much memory to give Operations Center software, it is necessary to try and determine the working set size.

While monitoring the `fsgc.log` file, look for any collection times that exceed 15 seconds or so; this signifies a potential problem. Follow the steps to the right to maintain consistently.

To determine Operations Center software's working set size:

1. Find the Server Java Runtime Option in Configuration Manager on the *Java Runtime* pane under *Tasks*.
2. Do one of the following to set verbose garbage collection logging:
   - On an IBM JVM, add `-Xverbosegclog:`*`OperationsCenter_install_path`*`/logs/gc.trc` to the value in the *Server Java Runtime Options* field.
   - On a Sun JVM, add `-Xloggc:`*`OperationsCenter_install_path`*`/logs/gc.trc` to the value in the *Server Java Runtime Options* field.
3. Set the -Xmx parameter to roughly one half of the machine's physical memory.
4. Click *Apply*, then close Configuration Manager.
5. View the `fsgc.log` file while running Operations Center to fully exercise the system to obtain the most accurate results.
6. Note that the VM is working to allocate memory, but might be having a difficult time.

   If the VM is performing many collections, one after the other (a few seconds apart, or even closer) then the VM is spending most of its time doing collection work.

   If you receive OutOfMemoryErrors go back to step 1 to give the Java VM more memory.

   Watch for lines labeled with Full GC. The final part of the line looks something like: 512245K->126256K (768000K). Ignore the first part of the line, which provides new generation collection information.

   The last part of the line indicates that the system started with 512245K, collected and compacted down to 126256K, and the heap size is currently 768000K. The middle number, in this case, 126256K, is the true working set size. It is the amount of memory that Operations Center software needs at its most compact and dense level.
7. Find the highest of the full GC "after" numbers, then add 40% and round up to the nearest 128M (use that value as the `-Xmx` parameter).

8. Set `-XX:MaxNewSize` to roughly one quarter of `-Xmx`, to a maximum of 512M.
9. Verify that the system runs well with the `mx` parameter.
10. Set `-Xms` to roughly one half of the `-Xmx` parameter.
11. If running in a 64-bit JVM, set the `-XX:MaxPermSize` parameter to `256m`.

Note that no operating system tools have been used to make these determinations.

On Solaris, use the `prstat` command to get a better view of memory usage. The most significant reason to do this is to see if there are other processes on the box using memory. Also use the Windows task manager to look at memory usage, but be careful because there is additional overhead included in the form of overhead of the VM.

The operating system tools can provide an idea of the amount of additional memory that the Java VM uses, over and above the amount provided to it with the `-Xmx` flag. The following is a sample `fsgc.log` output:

```
28.8972: [Full GC 4554K->4117K(5164K), 0.2139931 secs]
29.2526: [GC 4885K->4689K(7696K), 0.0138604 secs]
29.4526: [GC 5457K->5141K(7696K), 0.0220587 secs]
29.7094: [GC 5909K->5448K(7696K), 0.0169052 secs]
29.9499: [GC 6216K->5851K(7696K), 0.0221319 secs]
30.4152: [GC 6619K->6161K(7696K), 0.0179718 secs]
30.7133: [GC 6929K->6473K(7696K), 0.0180727 secs]
31.0125: [GC 7241K->6779K(7696K), 0.0182593 secs]
46.1367: [GC 7547K->6886K(7696K), 0.0117085 secs]
46.6746: [GC 7654K->6950K(7824K), 0.0089098 secs]
46.6837: [Full GC 6950K->6711K(7824K), 0.2935672 secs]
47.3671: [GC 7863K->6807K(12468K), 0.0049070 secs]
47.5725: [GC 7959K->6903K(12468K), 0.0055085 secs]
47.782: [GC 8055K->6997K(12468K), 0.0055513 secs]
48.0569: [GC 8149K->7093K(12468K), 0.0056057 secs]
48.4355: [GC 8245K->7188K(12468K), 0.0052199 secs]
78.6111: [Full GC 7830K->6281K(12468K), 0.2208297 secs]
```

The above example is from a small program run. Each Java VM produces different output, but the form should remain similar to this. Watch for the highest full GC number. In the above example, it is 6281K and the VM has a 12468K heap allocation size, at that point. Within the full GC line, additional information might be available and nested in brackets, such as:

```
78.6111: [Full GC [New 50k->16k(64K)] (7830K->6281K(12468K), 0.2208297 secs]
```

The total heap size (last bracketed number) should be the largest number.

The bottom line is:

1. Modify only the `-Xmx` and `-XX:MaxNewSize` flags at first.
2. `-XX:MaxNewSize` should be about one quarter, to a max of 512MB, of the `-Xmx` value.

Before varying from this, enlist some development support, then have at least two people study the `fsgc.log` file to determine if there is a problem that can be fixed with other flags.

6.1.6 Resolving Out of Memory Errors in Trace Logs

If an error message logged to the `formula.trc` file is similar to:

```
java.lang.OutOfMemoryError: PermGen space
```

then add the following JVM parameter for the Operations Center server:

`-XX:MaxPermSize=`*`size`*`m` Replace *`size`*

with a value greater than the default 128, where *size* represents that value. Note that the "m" is added to the value to represent megabytes.

For example:

```
-XX:MaxPermSize=128m -XX:MaxPermSize=256m
```

6.2 Configuring the Java Runtime Environment

Operations Center requires a Java Runtime Environment (JRE) and supports only specific versions of the JRE. For the current versions, see the *Operations Center 5.5 Getting Started Guide*. The JRE must be installed prior to installing Operations Center. For more information, see the *IOperations Center 5.5 Server Installation Guide*.

If the server is installed on a 64-bit VM, also install and configure an appropriate 64-bit JRE on the Operations Center server.

The JRE is set in Configuration Manager on the *Java Runtime* pane under *Tasks*.

7 Configuring and Administering the Database

Operations Center uses various embedded databases to store configuration data and support connections to both embedded and external data stores.

When configuring Operations Center, you'll need to set up and configure databases to handle the following features:

- **Configuration Storage:** define a database for Operations Center configuration data, version tracking, and to control connections to all databases configured for use with Operations Center, including the Event Data Store, Service Warehouse, Configuration Storage.
- **Event Store:** define a database for SNMP Integrator, Event Manager, and/or alarm suppression. Note that alarm suppression is not supported on Sybase.
- **Service Warehouse:** define a database to record alarm history and comments, historical performance and Service Level metrics.
- **Dashboard:** must be configured to use an external database. See the *Operations Center 5.5 Dashboard Guide* for more information and configuration information.

For information about supported databases and versions, see "Supported Databases" in the *Operations Center 5.5 Getting Started Guide*

Database Definitions are used to define database connections for Operations Center (except the Configuration Storage). They are also to connect to external data sources that provide additional data for Service Level Agreement calculations.

In addition, the following Operations Center features and products use embedded databases:

- **Configuration Storage:** initially uses an embedded data store before the external database is configured.
- **Dashboard:** is installed with a Hypersonic SQL data store for dashboard configuration settings. However, an external database must be configured for the dashboard.
- **SQL Views:** uses an Apache Derby database to make data available.

For information about embedded databases, see "Embedded Databases for Dashboard, SQL Views, and Operations Center Server" in the *Operations Center 5.5 Getting Started Guide*

The following sections provide further details regarding the various data stores and explain how to configure external databases for configuration storage, event store and the Service Warehouse:

- Section 7.1, "Configuring Windows Servers for Single Sign On (SSO)," on page 70
- Section 7.2, "Configuring the Database for Configuration Storage," on page 70
- Section 7.3, "Creating Database Definitions, Configuring the Event Data Store, and Connecting to External Databases," on page 75
- Section 7.4, "Viewing Database Status and Statistics," on page 89

- Section 7.5, “Configuring the Service Warehouse,” on page 90
- Section 7.6, “About Operations Center Embedded Databases,” on page 103

7.1 Configuring Windows Servers for Single Sign On (SSO)

Operations Center can be configured to use Server Domain Authentication for single sign on with Microsoft SQL databases. Windows servers require additional configuration for SSO functionality.

To configure Windows servers for Single Sign On with Operations Center MSSQL databases:

1. Stop the Operations Center server.
2. Copy one of the *`OperationsCenter_install_path`*`/mssql/win64/ntlmauth.dll` files into the directory containing the `java.exe` used by the Operations Center server.
3. If configuring Configuration Storage for Single Sign On, specify the domain, username and password when configuring the Configuration Storage definition.
   - On Windows servers, leave the username and password properties blank to attempt to connect using the current Windows user account.
   - If running Operations Center as a Windows service, the service must use the same domain user account as the database.

   For information on configuration storage database definition, see Configuring the Database for Configuration Storage, specifically Step 4 thru Step 9 on page 74.
4. Start the Operations Center server.
5. If configuring the Service Warehouse or other database definitions, specify the domain, username and password when configuring the Microsoft SQL database definition. See bulleted list in previous Step 3 for guidelines on specifying credentials and the domain.

   For information on Service Warehouse and other database definition, see Section 7.3.2, “Creating and Editing a Database Definition,” on page 86.

7.2 Configuring the Database for Configuration Storage

Configuration storage refers to stored configuration data in Operations Center as well as data used for version tracking. By default, the data store is an Object ODB that is embedded in Operations Center and installed with Operations Center. If there is an active database definition set up to use the Operations Center embedded database for Configuration Storage, the embedded database starts when the server starts.

However, configuring an external database for configuration storage is recommended. For more information on supported databases, see the *Operations Center 5.5 Getting Started Guide*. Note that not all of the databases support the Version Tracking functionality. For more information, see the *Operations Center 5.5 Version Tracking Guide*.

The following sections provide details on configuring the Configuration Storage database and enabling processes to perform maintenance on database entries:

- Section 7.2.1, “Configuring the Database,” on page 71
- Section 7.2.2, “Specifying a Configuration Storage Database,” on page 72
- Section 7.2.3, “Configuring Maintenance Options for the Configuration Storage Database,” on page 74

7.2.1 Configuring the Database

The following databases require configurations in order to use them for configuration storage:

- "General Database Requirements" on page 71
- "DB2" on page 71
- "Oracle 10g" on page 71
- "Oracle RAC" on page 71

General Database Requirements

The following database requirements apply when configuring a database for configuration storage:

- Verify you have set the proper log space requirements as advised by the appropriate database vendor.
- Database must be case insensitive.

DB2

The database default bufferpool and table space size must be 8K or larger when using DB2 as the Configuration Storage database.

Be sure to create a user temporary tablespace and ensure that the proper access rights granted to the database user.

Oracle 10g

If using Oracle 10g as the database for Configuration Storage, set the database character set to the default value. The default character set is based on language settings from the operating system.

The following errors have occurred in the `formula.trc` file when using Oracle 10g with the character setting of `Unicode (AL32UTF8).`

```
WARN org.hibernate.util.JDBCExceptionReporter - SQL Error: 24813, SQLState: 99999

ERROR org.hibernate.util.JDBCExceptionReporter - ORA-24813: cannot send or receive
an unsupported LOB

WARN Adapter.Meta.Graphics - Unable to load graphics setup
```

Oracle RAC

To configure Oracle RAC for Configuration Storage:

1. Download and install the Oracle Instant Client (http://www.oracle.com/technetwork/database/features/instant-client/index-097480.html).
2. Copy the `ojdbc6.jar` file from the Oracle Instant Client installataion to your `jre/lib/ext` directory.
3. Create a file named `tnsnames.ora` in the Oracle Instant Client installataion directory, then add the following entry to the file:

   ```
   racdb = (DESCRIPTION=
   (ADDRESS=(PROTOCOL=TCP)(HOST=node1_ipAddress)(PORT=node1_portNumber))
   ```

```
(ADDRESS=(PROTOCOL=TCP)(HOST=node1_ipAddress)(PORT=node1_portNumber))
(ADDRESS=(PROTOCOL=TCP)(HOST=node2_ipAddress)(PORT=node2_portNumber))
(LOAD_BALANCE=yes)
(CONNECT_DATA=(SERVER=DEDICATED)(SERVICE_NAME=rac_service_name)
(FAILOVER_MODE=(TYPE=SELECT)(METHOD=BASIC)(RETRIES=180)(DELAY=5))))
```

4 To allow the OCI `dll` files to be loaded, edit the system environment variables:

- Add an entry for `TNS_ADMIN` with the value of `drive:/OracleInstantClient_install_path`
- Edit the `LIB` and `PATH` variables to add `drive:/OracleInstantClient_install_path` to the *beginning* of both strings.

Then, verify the changes to the system environment variables have been applied.

5 Continue to Section 7.2.2, "Specifying a Configuration Storage Database," on page 72.

Select *Oracle Database Server (OCI/Native Drivers)* as the configuration storage type to connect to RAC.

7.2.2 Specifying a Configuration Storage Database

The Configuration Storage database is not configured using a Database Definition, but through the Configuration Manager (in the *Server* pane) instead. For more detailed information about the Configuration Manger, see Updating Server Settings Using the Configuration Manager (page 17). Configuration Storage data can be shared among multiple Operations Center servers. For more information on configurations, see Chapter 8, "Managing Configurations," on page 105.

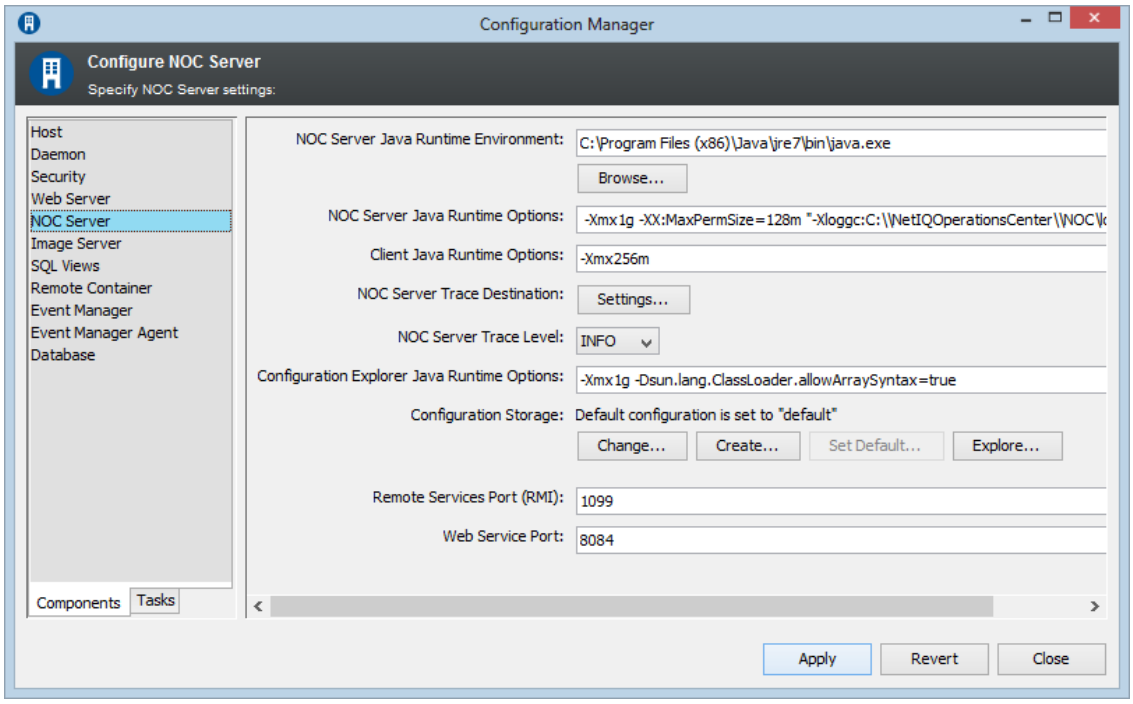
To specify a configuration storage database:

1 Perform any steps necessary to configure the database. For instructions, see Section 7.2.1, "Configuring the Database," on page 71.

2 Access the Configuration Manager.

For information on launching the Configuration Manager, see Section 2.1, "Accessing and Using Configuration Manager," on page 17.

3 On the *Components* tab of Configuration Manager, click *NOC Server*.

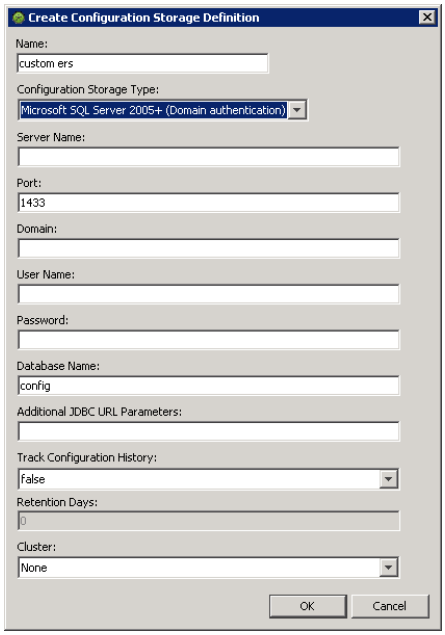


4 Do one of the following for the *Configuration Storage*:

- Click *Create* to create a new Configuration Storage definition.

  The *Create Configuration Storage* dialog box opens.
- Click *Change* to change the parameter for a Configuration Storage.
- If you have multiple Configuration Storage data stores, click *Set Default*, then select one from the list.

5 Enter a name for the new configuration in the *Name* field.



6 From the *Configuration Storage Type* list, select the database type.

Properties fields display based on the type selected.

7 Specify the required properties to connect to the database (properties vary depending on the database type selected):

Server Name: The name of the database server.

Port: The port number used by the database server to communicate with Operations Center. A default port is provided for each type of database.

Domain: The domain to use for domain authentication with single sign on (used when defining a Microsoft SQL Server database with Domain Authentication).

User Name: The User ID with administrator rights to access the database.

If using Microsoft SQL Server with Domain Authentication, specify the username of the Windows user account, or leave blank (on Windows servers) to attempt to use the credentials of the currently active Windows user account.

Password: The password for the user account entered above.

If using Microsoft SQL Server with Domain Authentication, specify the password of the Windows user account, or leave blank (on Windows servers) to attempt to use the credentials of the currently active Windows user account.

Database Name: Name of the database to be used as the Configuration Storage data store.

Additional JDBC URL Parameters: (Optional) Additional JDBC URL property values for Microsoft SQL Server and Microsoft SQL Server (Domain Authentication) databases. Prefix each parameter entry with a semi-colon, such as:

`;parameter1=value1;parameter2=value2`

For example,

`;progName=NOC;domain=mosol`

SID: The System ID for an Oracle database.

Service Name: The service name for an Oracle RAC database.

Track Configuration History: Enables Version Tracking. For more information on configuring Version Tracking, see *Operations Center 5.5 Version Tracking Guide*.

Retention Days: The number of days to retain version tracking data. For more information on configuring Version Tracking, see *Operations Center 5.5 Version Tracking Guide*.

8 Leave *None* in the *Cluster* drop-down unless the Operations Center server is part of a clustered environment. For information and requirements about configurations in a clustered environment, see "Implementing a High Availability Solution" in the *Operations Center 5.5 Server Installation Guide*.

9 Click *OK*.

7.2.3 Configuring Maintenance Options for the Configuration Storage Database

A clean up process can be enabled that performs a maintenance clean up on Configation Storage database entries when the Operations Center server starts. Running the clean up process removes entries for elements under `root=Elements` that do not have relationships, attributes, or children.

In order for the clean up process to successfully run:

- Version Tracking must not be enabled.

For more information about Version Tracking, see *Operations Center 5.5 Version Tracking Guide*.

- The server must be specified as the primary database writer (or cluster coordinator) if the server is part of a clustered environment.

  For more information on configuring Operations Center servers in a clustered environment, see "Configuring the Operations Center Server, Database, and Data Warehouse" in the *Operations Center 5.5 Server Installation Guide*.

To enable ConfigStore clean up:

1. Open the */OperationsCenter_install_path*/config/Formula.custom.properties file (or create one if you do not already have it) to add or edit the following properties:
   - **Server.cleanConfigEntries.atServerStart** Set to `true` to start the clean up of configuration entries when the server starts.
   - **Server.cleanConfigEntries.maxTraverseDepth** Specify the maximum number of levels to traverse for elements under `root=Elements`. Set to `0` to not impose any limit.
   - **Server.cleanConfigEntries.maxTraverseEntries** Specify the maximum number of entries to traverse for elements under `root=Elements`. Set to `0` to not impose any limit.
   - **Server.cleanConfigEntries.maxRemoveEntries** Specify the maximum number of qualifying entries to remove for elements under `root=Elements`. Set to `0` to not impose any limit.

   Note that these properties can be copied from the */OperationsCenter_install_path*/config/formula.properties into the Formula.custom.properties files, then uncommented and customized.

   For more information about using the Formula.custom.properties file to customize configuration options, see Section 2.4, "Making Custom Changes," on page 37.
2. Stop and restart the Operations Center server for the changes to take effect.

7.3 Creating Database Definitions, Configuring the Event Data Store, and Connecting to External Databases

A database definition defines database settings required to establish a connection between Operations Center and a database that will be used by Operations Center for one or more of the following:

- Event Data Store. Create a database definition for the Event Data Store only if you are using Event Manager, SNMP Integrator, or alarm suppression (note that alarm suppression is not supported on Sybase).

  For more information about these products, see the *Operations Center 5.5 Event Manager Guide* or the *Operations Center 5.5 SNMP Integrator Guide*. For alarm suppression topics, see Section 11.6, "Configuring Suppression and Acknowledgement," on page 157.
- Service Warehouse. See Section 7.5, "Configuring the Service Warehouse," on page 90 for specific steps to create the Service Warehouse.
- External Data Source for service level agreement (SLA) management in Operations Center. For more information about configuring and using an external data source for this purpose, see the *Operations Center 5.5 Service Level Agreement Guide*.

The following sections cover the steps necessary to setup and maintain a connection to an external database using a database definition:

- Section 7.3.1, "Setting Up an External Database," on page 76
- Section 7.3.2, "Creating and Editing a Database Definition," on page 86
- Section 7.3.3, "Enabling a Definition," on page 88
- Section 7.3.4, "Initializing a Definition," on page 88
- Section 7.3.5, "Disabling and Deleting a Definition," on page 89

7.3.1 Setting Up an External Database

- "Understanding General Requirements and Configurations" on page 76
- "Reviewing and Modifying Database Scripts" on page 77
- "Configuring DB2" on page 77
- "Configuring MSSQL" on page 78
- "Configuring Oracle" on page 80
- "Oracle RAC" on page 82
- "Configuring PostgreSQL" on page 83
- "Configuring Sybase" on page 83

Understanding General Requirements and Configurations

For database requirements, see "Databases" in the *Operations Center 5.5 Getting Started Guide*.

The following are general suggestions when setting up and configuring your external database for the Service Warehouse or the Event Data Store:

- Create a `formula` user with appropriate `create/drop/alter` object permissions. This allows the Operations Center server to create the required database tables for the Service Warehouse.
- Assign a separate default tablespace to the `formula` user instead of using the standard `user` tablespace. This isolate Operations Center from any existing data sources and provide a container for all performance and SLA data which can be managed separately.
- The default tablespace for the `formula` database user should have enough space available to create the necessary tables and maintain at least 1 weeks worth of performance/alarm history data (approximately 200 MB).
- Configure your database to handle a substantial amount of data. The `CreateFormula.sql` script provided for each database type creates a default user and tablespace settings for small Operations Center configurations. To build a database for a large configuration, consult the documentation provided by the database vendor. For additional sizing information, see Section 7.5.1, "Sizing the Service Warehouse," on page 91.
- The schema is the type of data store. Define multiple schemas for a single database. When multiple schemas exist within the same database, a unique ID must be created for each schema. If you select multiple schemas for a single definition, do not select a schema that is already enabled in another database definition.
- If auditing is enabled for your database, be aware that failed inserts to the `BSAAlarmElements` table will occur under some circumstances. Please disable logging of failed inserts to this table as it could negatively affect database performance.

Reviewing and Modifying Database Scripts

Operations Center ships with sample scripts for configuring an instance of each database type. These scripts are located in the appropriate database's subdirectory under `/OperationsCenter_install_path/databases/samples`. The directories for sample databases are: dB2, Oracle, MSSQL (for Microsoft SQL Server), PostgreSQL, and Sybase.

A database administrator should review and modify these scripts to ensure creation of an appropriate database for the corporate infrastructure.

IMPORTANT: Before creating a database, refer to the subsequent section below on each specific database type for the information about setting up the specific database.

Configuring DB2

Refer to the following sections for steps on configuring an Operations Center database using a DB2 database:

- "DB2 Requirements" on page 77
- "Installing Generic JDBC Drivers" on page 77
- "Running the Operations Center Script to Create the Database Instance" on page 78

DB2 Requirements

The database default bufferpool and table space size must be 16K or larger when DB2 is used as the Service Warehouse database.

Installing Generic JDBC Drivers

The drivers included with Operations Center do not support DB2. The generic JDBC drivers supplied with the DB2 distribution must be used. These drivers can be found in `/DB2_root_directory/java`.

To install the jdbc 2.0 drivers into the Operations Center installation:

1. Verify your DB2 distribution is using the jdbc 2.0 drivers.
   - **1a** Shutdown your DB2 server (including all services).
   - **1b** Change directory to `/DB2_root_directory/java12`.
   - **1c** Run the usejdbc20.bat script.
   - **1d** Restart the DB2 server (including all services).
2. Configure the DB2 database properties using the Operations Center Configuration Manager by entering the path to the DB2 driver archive in the properties panel.

   For example, `/DB2_root_directory/java/db2java.zip`.

   The Configuration Manager reads the archive and re-writes it to `/OperationsCenter_install_path/classes/ext` directory.

   This step is important since there are cases where the default IBM archive format is incompatible with the JRE distribution supplied with Operations Center.
3. Restart the Operation Center server.

Running the Operations Center Script to Create the Database Instance

The Operations Center script used to create the sample Service Warehouse is '`CreateFormula.sql`. The script must be executed by a user with *admin* privileges since it creates a new database, a bufferpool compatible with the Operations Center schema, and a sample tablespace for the Service Warehouse.

To run the script to create the database instance:

1. Edit the `/OperationsCenter_install_path/database/samples/db2/CreateFormula.sql` script to customize the database instance parameters:
   - Update the installation location to a suitable directory.

     The default directory for the database and tablespaces is `/OperationsCenter_install_path/databases/samples/db2`.
   - Confirm the size for each data file.

     Search on the *FILE* string to review each occurance.
   - Update to create new buffer pools as necessary. Examples are provided in the script.
2. Run the `CreateFormula.sql` script by issuing the following command:

   ```
   db2cmd -c db2 -t -z create.log -f CreateFormula.sql
   ```

   This script requires *admin* privileges.
3. After the DB2 database instance is created, create a new *Database Defintion* for the database instance.

   The default value for the *Listener Port* property is *50000*. If you setup DB2 to listen on a different port, specify the new port in this property. See the DB2 documentation for instructions on how to change the default listener port.

   For more information on creating database definitions, see Section 7.3.2, "Creating and Editing a Database Definition," on page 86.
4. If you did not enable the database definition as a part of Step 3, right-click the database definition and select *Enable Database Definition*.
5. If the database definition is for a single schema for *Event Data Store*, it is necessary to manually initialize the schema. Right-click the database definition, then select *Initialize Database Schema*.

   Normally, this step is not necessary since the Operations Center server creates all appropriate schemas when required.

Configuring MSSQL

Refer to the following sections for steps on configuring an Operations Center database using a MSSQL database:

- "Requirements for MSSQL Server Installation" on page 78
- "Running the Operations Center Script to Create the Database Instance" on page 79

Requirements for MSSQL Server Installation

While installing SQL Server, select *Mixed Mode* when specifying the security mode (authentication). Mixed Mode enables users to connect using Windows Authentication or SQL Server Authentication. Users who connect through a Microsoft Windows user account can use trusted connections (connections validated by Windows) in either Windows Authentication Mode or Mixed Mode. SQL Server Authentication is provided for backward compatibility.

Running the Operations Center Script to Create the Database Instance

The Operations Center script used to create the sample Service Warehouse is 'CreateFormula.sql. The script must be executed by a user with *dba* privileges since it creates the default database and user.

To run the script to create the database instance:

1. Edit the */OperationsCenter_install_path*/database/samples/mssql/CreateFormula.sql script to customize the database instance parameters:
   - Update the *dataDir* variable to point to a pre-existing directory outside the Operations Center installation directory.

     ```
     SET @dataDir = 'd:/OperationsCenter_install_path/database/mssql'
     ```

     The default directory for the database and tablespaces is */OperationsCenter_install_path*/databases/samples/mssql.

     The specified directory must exist prior to executing the script.
   - Configure the initial and maximum tablespace sizes. The defaults specify an initial database size of 200 MB and a maximum size of 2000. Change these values if required.

     ```
     SET @initialSize = 200
     ```

     ```
     SET @maximumSize = 2000
     ```

     These values depend on available disk space and any operating system imposed file size limits.
   - The default database user and password created by CreateFormula.sql are *formula* and *sesame* respectively. If you prefer to use different values, change them by resetting the following variables:

     ```
     SET @username = N'formula'
     ```

     ```
     SET @password = N'sesame'
     ```
2. Run the CreateFormula.sql script by issuing one of the following commands:
   - If using isql.exe:

     ```
     isql -n -U sa -P your-sa-password -S your-server-name -i CreateFormula.sql
     ```
   - If using Microsoft's SQL Query Analyzer:
     1. Launch the *SQL Server Enterprise Manager* from the *Microsoft SQL Server* program menu.
     2. Click *Tools* > *SQL Query Analyzer*.
     3. Click *File* and select *Open*, then browse to the location of the CreateFormula.sql script and select it.
     4. Click *Tools* and select *Execute*.

        The new Operations Center database is created.

   This script requires *dba* privileges.
3. After the MSSQL database instance is created, create a new *Database Definition* for the database instance.

   The default value for the *Listener Port* property is *1433*. If you setup SQL Server to listen on a different port, specify the new port in this property. See the SQL Server documentation for instructions on how to change the default listener port.

   For more information on creating database definitions, see Section 7.3.2, "Creating and Editing a Database Definition," on page 86.

4 If you did not enable the database definition as a part of Step 3, right-click the database definition and select *Enable Database Definition*.

5 If the database definition is for a single schema for *Event Data Store*, it is necessary to manually initialize the schema. Right-click the database definition, then select *Initialize Database Schema*.

Normally, this step is not necessary since the Operations Center server creates all appropriate schemas when required.

Configuring Oracle

Refer to the following sections for steps on configuring an Operations Center database using an Oracle database:

- "Requirements for Oracle Database Installation" on page 80
- "Requirements for Database User Accounts, Table Spaces and Privileges" on page 80
- "Running the Operations Center Script to Create the Database Instance" on page 81

Requirements for Oracle Database Installation

Check your current maximum number of connections parameter in the Service Warehouse database connection properties.The following properties must be set in the init<SID>.ora or init.ora in order for the Operations Center database connection pool to operate correctly:

- `processes = 200`
- `open_cursors = 500`

If Operations Center is configured to use more than 10 database connections, the 'processes' and 'open_cursors' properties should be adjusted as follows:

- `processes: number-of-connections * 20`
- `open_cursors: number-of-connections * 50`

The default rollback segment is usually too small for the Service Warehouse purge process to work. We recommend increasing it by * 4 to start, then monitor and adjust as the warehouse gets larger and starts purging old records as defined by the profiles.

Requirements for Database User Accounts, Table Spaces and Privileges

One Oracle database instance is sufficient for Operations Center's database requirements, but it is necessary to configure 4 different user accounts pointing to dedicated individual table spaces.

Add IDs and table spaces for Configstore, Service Warehouse, Event Data Store and Dashboard. The user privileges are the same for all databases, except Configstore which needs additional privileges. All table spaces must have auto extend enabled.

The following are basic examples to illustrate and understand the requirement:

```
User ID: Dashboard_username
Tablespace: DASHBOARD_FILESPACE
Expected Size: <5 GB
Role: APPLICATION_DEVELOPER
Privileges: CREATE SESSION and CREATE TABLE
```

User ID: *Warehouse_username*
Tablespace: WAREHOUSE_FILESPACE
Expected Size: Unknown but could be very large with hundreds of GBs depending on your profiles and volume.
Role: APPLICATION_DEVELOPER
Privileges: CREATE SESSION and CREATE TABLE

User ID: *EventDataStore_username*
Tablespace: EVENTDATASTORE_FILESPACE
Expected Size: <5 GB
Role: APPLICATION_DEVELOPER
Privileges: CREATE SESSION and CREATE TABLE

User ID:*Configstore_username*
Tablespace: CONFIGSTORE_FILESPACE
Expected Size: <5 GB
Role: APPLICATION_DEVELOPER
Privileges: CREATE SESSION, CREATE TABLE, CREATE SEQUENCE and CREATE TRIGGER

Running the Operations Center Script to Create the Database Instance

The Operations Center script used to create the sample Service Warehouse is 'CreateFormula.sql. The script must be executed by an *Oracle* system user since it creates a default role, user, and a set of sample tablespace data files for the Service Warehouse.

To run the script to create the database instance:

1. Verify you have a database instance setup for the Service Warehouse.

   To setup a special instance, use Oracle's *Database Configuration Assistant.*

2. Edit the */OperationsCenter_install_path*/database/samples/oracle/CreateFormula.sql script to customize the database instance parameters:
   - Change the *userID*, *passwd*,' and *roleName* variables if the defaults are not acceptable. The defaults are:
     - DEFINE userID = formula
     - DEFINE passwd = sesame
     - DEFINE roleName = formula_role
   - Customize the installation location by updating the *dataDir* variables.

     The default directory for the Operations Center database and tablespace data is $ORACLE_HOME/oradata.

     If the tablespaces are to be spread across multiple filesystems, set each variable individually. Otherwise, leave the dataDir[2-4] variables set to the defaults:
     - DEFINE dataDir1 = /fs01/oradata/FORMULA
     - DEFINE dataDir2 = /fs02/oradata/FORMULA
     - DEFINE dataDir3 = &dataDir1
     - DEFINE dataDir4 = &dataDir2

     The specified directory must exist and be owned by the *Oracle* user prior to executing the script..
   - Configure the initial and maximum tablespace sizes if necessary.

The defaults specify an initial tablespace size of 200M and a maximum tablespace of 2GB:

- `DEFINE initialSize = 200M`
- `DEFINE maximumSize = 2000M`

These values depend on available disk space and any operating system imposed file size limits. Because the *AUTOEXTEND* feature is enabled on all Operations Center tablespaces, the maximum size variable should be set.

3 Run the *CreateFormula.sql* script by issuing the following command:

```
sqlplus system
Password: your-system-password
SQL> @CreateFormula.sql
SQL> exit;
```

This script requires *oracle system* privileges.

4 After the Oracle database instance is created, create a new Database Defintion for the database instance.

The default value for the *Listener Port* property is *1521*. If you setup Oracle to listen on a different port, specify the new port in this property. See the Oracle documentation for instructions on how to change the default listener port.

For more information on creating database definitions, see Section 7.3.2, "Creating and Editing a Database Definition," on page 86.

5 If you did not enable the database definition as a part of Step 3, right-click the database definition and select *Enable Database Definition*.

6 If the database definition is for a single schema for *Event Data Store*, it is necessary to manually initialize the schema. Right-click the database definition, then select *Initialize Database Schema*.

Normally, this step is not necessary since the Operations Center server creates all appropriate schemas when required.

Oracle RAC

Refer to the following sections for steps on configuring an Operations Center database using an Oracle RAC:

- "Configuring the Oracle RAC" on page 82
- "Running the Operations Center Script to Create the Database Instance" on page 82

Configuring the Oracle RAC

If using Oracle RAC as the database for the Service Warehouse, a few configurations are required. These configurations are the same as when you configure Oracle RAC for configuration storage. For instructions, see "Oracle RAC" on page 71.

Running the Operations Center Script to Create the Database Instance

The Operations Center script used to create the sample Service Warehouse is `'CreateFormula.sql`. The script must be executed by an *Oracle* system user since it creates a default role, user, and a set of sample tablespace data files for the Service Warehouse.

For more information on running the `CreateFormula.sql` on Oracle databases, see "Running the Operations Center Script to Create the Database Instance" on page 81.

Configuring PostgreSQL

Refer to the following sections for steps on configuring an Operations Center database using a PostgreSQL database:

- "Running the Operations Center Script to Create the Database Instance" on page 83

Running the Operations Center Script to Create the Database Instance

The Operations Center script used to create the sample Service Warehouse is `'CreateFormula.sql`.

To run the script to create the database instance:

1. Edit the `/OperationsCenter_install_path/database/samples/postgresql/CreateFormula.sql` script to customize the database instance parameters:
   - Change the *userID* and *passwd* variables if the defaults are not acceptable. You must change the *location* variable in the `CREATE TABLESPACE` command as it must point to an existing directory to which the PostgreSQL server can modify. The defaults are:
     - `CREATE USER formula PASSWORD 'formula' ;`
     - `CREATE TABLESPACE BSAWarehouse OWNER formula LOCATION '/var/lib/pgsql/data/BSAWarehouse';`
     - `CREATE DATABASE formula WITH OWNER formula TABLESPACE BSAWarehouse ;`
2. Run the *CreateFormula.sql* script by issuing the following command:

   `psql -U admin_account -f CreateFormula.sql`
3. After the PostgreSQL database instance is created, create a new Database Defintion for the database instance.

   The default value for the *Listener Port* property is *5432*. If you setup PostgreSQL to listen on a different port, specify the new port in this property. See the PostgreSQL documentation for instructions on how to change the default listener port.

   For more information on creating database definitions, see Section 7.3.2, "Creating and Editing a Database Definition," on page 86.
4. If you did not enable the database definition as a part of Step 3, right-click the database definition and select *Enable Database Definition*.
5. If the database definition is for a single schema for *Event Data Store*, it is necessary to manually initialize the schema. Right-click the database definition, then select *Initialize Database Schema*.

   Normally, this step is not necessary since the Operations Center server creates all appropriate schemas when required.

Configuring Sybase

Refer to the following sections for steps on configuring an Operations Center database using a Sybase database:

- "Requirements for Sybase Server Installation" on page 84
- "Sybase Configuration Requirements" on page 84
- "Running the Operations Center Script to Create the Database Devices and Database Instance" on page 84
- "Increasing Maximum Locks to Resolve Deadlock Issues with Sybase" on page 85

Requirements for Sybase Server Installation

The scripts in the sample Sybase directory create a moderate database consisting of a 200MB Operations Center log device and a 100M data device. Also, since the Service Warehouse uses a substantial amount of temp space for sorting during large queries, a new 50MB device for the tempdb database is created.

To support JDBC DatabaseMetaData methods, Sybase provides a set of stored procedures that jConnect calls for metadata about a database. These stored procedures must be installed on the Sybase server for the JDBC metadata methods to work. If the stored procedures for providing metadata are not already installed on your Sybase server, install them using the stored procedure scripts provided with jConnect. For complete instructions on installing stored procedures, see the *Sybase jConnect for JDBC Installation Guide* and *Release Bulletin* from your Sybase installation or go to Sybase's online documentation for installing stored procedures (http://infocenter.sybase.com/help/index.jsp?topic=/com.sybase.infocenter.dc32179.0700/html/jconnig/CJHIEFJF.htm).

Sybase Configuration Requirements

The following server configuration changes are required:

- **max_network_packet_size:** set to `16384`. Default is `2048`.
- **procedure_cache_size:** set to `256000`. Default is `7000`.
- **statement_cache_size:** set to `256000`. Default is `7000`.
- **user_log_cache_size:** set to `524288`. Default is `4096`.

The following PostgreSQL database configuration changes are required:

- **ddl_in_tran:** set to `true`.

Running the Operations Center Script to Create the Database Devices and Database Instance

The Operations Center script used to create the sample Service Warehouse database devices is `FormulaDisk.sql`. This script must be executed by a user with *dba* privileges since it creates the Operations Center disk devices. The Operations Center script used to create the sample Service Warehouse is `CreateFormula.sql`. The script must be executed by a user with *dba* privileges since it creates both the default database and user.

To create the database devices and instance on a Sybase database:

1. Edit the */OperationsCenter_install_path*`/database/samples/sybase/FormulaDisk.sql` script to customize the database devices parameters:

   Before running FormulaDisk.sql, you must change the location and size of the device data files to point to a desired directory. Note also, the specified directory must exist prior to executing the FormulaDisk.sql script.

   - Edit the following variable to change the name of the database.

     ```
     SELECT @dbName = "formula"
     ```

   - Change the location and size of the device data files for both the `formulaDisk` and `formulaLog` devices if necessary:

     ```
     SELECT @dskDev  = "/opt/sybase/dbs/formulaLog.dat"
     SELECT @dskSize = 1024000    /* approx 200M */
     SELECT @logDev  = "/opt/sybase/dbs/formulaDsk.dat"
     SELECT @logSize = 51200      /* approx 100M */
     ```

   - Configure the initial and maximum database size variables.

These values depend on available disk space and any operating system imposed file size limits.

- Update to create new buffer pools as necessary. Examples are provided in the script.

2 Edit the */OperationsCenter_install_path*/database/samples/sybase/CreateFormula.sql script to customize the database instance parameters:

- Change the database name, user login, and password settings if the defaults are not desirable. The dbName must correspond to value of @dbName set in FormulaDisk.sql.

```
SELECT @dbName   = "formula".
SELECT @userName = "formula"
SELECT @password = "sesame"
```

3 Run the *FormulaDisk.sql* script by issuing the following command:

```
isql -Usa -Psa-password -Sserver-name -i FormulaDisk.sql
```

Where, *sa-password* is the Sybase system administrator password and *server-name* is the Sybase server name.

4 Run the *CreateFormula.sql* script by issuing the following command:

```
isql -Usa -Psa-password -Sserver-name -i CreateFormula.sql
```

Where, *sa-password* is the Sybase system administrator password and *server-name* is the Sybase server name.

5 After the Sybase database instance is created, create a new Database Defintion for the database instance.

The default value for the *Listener Port* property is *4100*. If you setup Sybase to listen on a different port, specify the new port in this property. See the Sybase documentation for instructions on how to change the default listener port.

For more information on creating database definitions, see Section 7.3.2, "Creating and Editing a Database Definition," on page 86.

6 If you did not enable the database definition as a part of Step 3, right-click the database definition and select *Enable Database Definition*.

7 If the database definition is for a single schema for *Event Data Store*, it is necessary to manually initialize the schema. Right-click the database definition, then select *Initialize Database Schema*.

Normally, this step is not necessary since the Operations Center server creates all appropriate schemas when required.

Increasing Maximum Locks to Resolve Deadlock Issues with Sybase

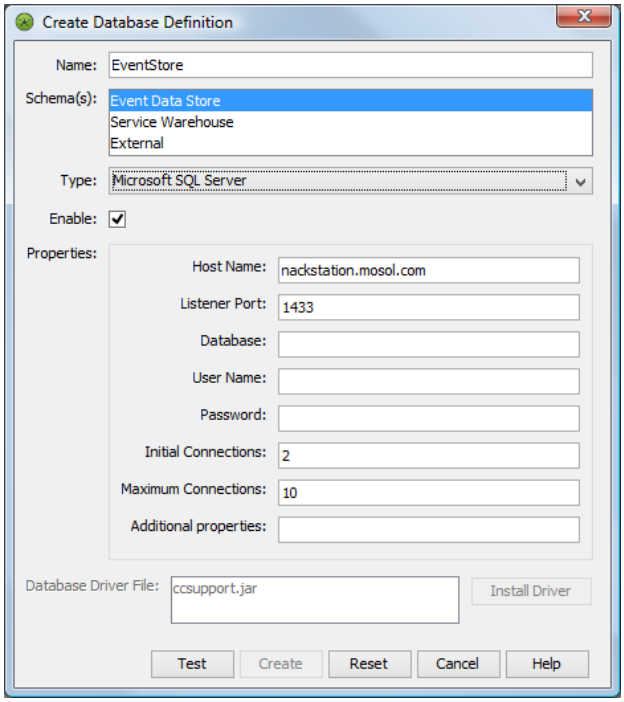
To attempt to deal with deadlock issues in a Sybase environment, the database tables created by the data warehouse use the "row-level" locking feature. This reduces the number of lock collisions in a fairly busy system. In order to accommodate the table setting, the number of locks that can be held open by the system can be re-configured by the CreateFormula.sql script. Note that changing this parameter requires the Sybase server to be stopped and restarted.

7.3.2 Creating and Editing a Database Definition

Databases definitions are created and edited in the Operations Center console. Each database connection appears as an element in the hierarchy under *Enterprise* > *Administration* > *Database Definitions*.

To create a database definition:

1. In the Operations Center console *Explorer* pane, expand the *Administration* root element.
2. Right-click the *Database Definitions* element, then select *Create Database Definition* to open the Create Database Definition dialog box.



3. Enter a name for the definition in the *Name* field.

   It is best practice to use a name without spaces or special characters.
4. In the Schema(s) section, select one or more database schemas to indicate the type of data store.

   Use the Ctrl or Shift keys to select multiple schemas.

   If you select multiple schemas, do not select a schema that is already enabled in another database definition.
5. Click the *Type* list, then select a database type.

   If selecting *(SQL Server Domain Authentication)*, there are required configurations for Windows servers. For more information, see Section 7.1, "Configuring Windows Servers for Single Sign On (SSO)," on page 70.

   Select *Other* if you are using a database type that is not supported by Operations Center.
6. Deselect the *Enable* check box if you do not wish to activate the database definition. If the database is not enabled on create, then it needs to be manually enabled later. See Section 7.3.3, "Enabling a Definition," on page 88.

7 Specify the required properties to connect to the database (properties vary depending on the database type selected):

Database Properties	Description
Hostname	The name of the database server.
Listener Port	The port on which the database listens for communications.
Server ID (SID)	The name of the database (used when defining an Oracle database connection).
Database	The name of the database (used when defining a Microsoft SQL Server, Sybase, or DB2 database connection).
Domain	The domain to use for domain authentication with single sign on (used when defining a Microsoft SQL Server database with Domain Authentication).
User Name	The name of the user account. When multiple Operations Center schemas exist within the same database, a unique ID must be created for each database schema. If using Microsoft SQL Server with Domain Authentication, specify the username of the Windows user account, or leave blank (on Windows servers) to attempt to use the credentials of the currently active Windows user account.
Password	The password for the user account. If using Microsoft SQL Server with Domain Authentication, specify the password of the Windows user account, or leave blank (on Windows servers) to attempt to use the credentials of the currently active Windows user account.
Initial Connections	The number of connections established upon initial connection.
Maximum Connections	The maximum number of connections allowed.

Additional Properties	(Optional) Additional JDBC URL properties for Microsoft SQL Server and Microsoft SQL Server (Domain Authentication) databases. Prefix each parameter entry with a semi-colon, such as:
	
	`;parameter1=value1;parameter2=value2`
	
	For example,
	
	`;progName=NOC;domain=mosol`

Database Class, URL, and Driver File	Name and location of the IBM DB2 driver. If using an IBM DB2 database, it is necessary to upload the JDBC driver file supplied by IBM. Contact Support (https://www.netiq.com/support/) to have the driver signed by Operations Center. Click the *Install Driver* button to browse, then select a driver file.

8 If you are configuring an non-supported database and selected *Other* in Step 5 on page 86, click *Install Driver* to install a custom driver.

Because of the tightened security model introduced by Oracle Java 7 Update 45, custom drivers must be signed by Operations Center. Contact Support (https://www.netiq.com/support/) to have the driver signed.

9 Click the *Test* button to test and verify that the database settings are valid, prior to creating the definition.

If the database connection is valid, the *Create* button activates.

10 Click the *Create* button to create and save the database definition.

The new definition displays under the *Database Definitions* element in the hierarchy.

11 If the database definition is only using the Event Data Store schema, then it must be initialized. Continue to Section 7.3.4, "Initializing a Definition," on page 88. If created in conjunction with the Service Warehouse schema, this step is not necessary.

To edit an existing definition, right-click the definition, then select *Properties*. On the Properties dialog box, select *Database*. The options are the same as the *Create a Database* dialog box.

7.3.3 Enabling a Definition

After creating a database definition, it must be enabled. Using default settings, Database Definitions are enabled on creation.

To manually enable a database definition:

1 In the Operations Center console *Explorer* pane, expand the *Administration* root element.

2 Right-click the database element under *Database Definitions*, then select *Enable Database Definition*.

7.3.4 Initializing a Definition

Initializing is not usually necessary after enabling a database definition since the Operations Center server creates all appropriate schemas when required. However, if creating a database definition with a single schema for Event Data Store, the schema must be manually initialized after creating the definition.

Any of the databases defined using a database definition can be reinitialized. However, reinitializing a database drops the entire database schema and all the data before recreating the schema.

WARNING: Use extreme caution when reinitializing a database because it drops the entire database schema, including all data, to re-create the schema.

To initialize or reinitialize a database:

1. In the *Explorer* pane, expand the *Database Definitions* element.
2. If the database definition is not enabled, right-click the database definition and select *Enable Database Definition*.
3. Right-click the database definition, then select *Initialize Database Schema*.

7.3.5 Disabling and Deleting a Definition

A database definition can be disabled or deleted. Disabling a definition retains the definition but the connection is no longer active.

To manually disable or delete a database definition:

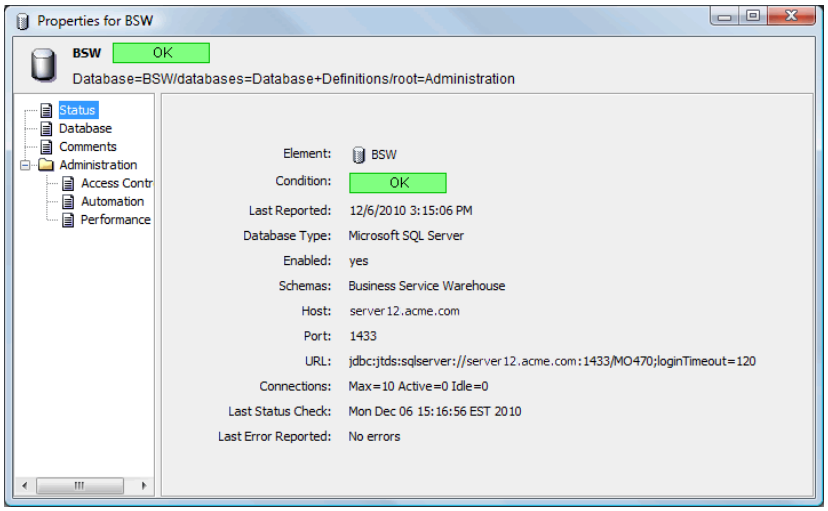
1. In the Operations Center console *Explorer* pane, expand the *Administration* root element.
2. Right-click the database element under *Database Definitions*, then select one of the following: select *Enable Database Definition*.
   - *Disable Database Definition* to disable the database definition.
   - *Delete Database Definition* to delete the database definition.

7.4 Viewing Database Status and Statistics

After a database definition has been created, including the Event Data Store and the Service Warehouse, check the status by viewing the properties of the database element. Also, view the status of all databases by viewing the properties of the *Database Definitions* element.

To view the status of a database:

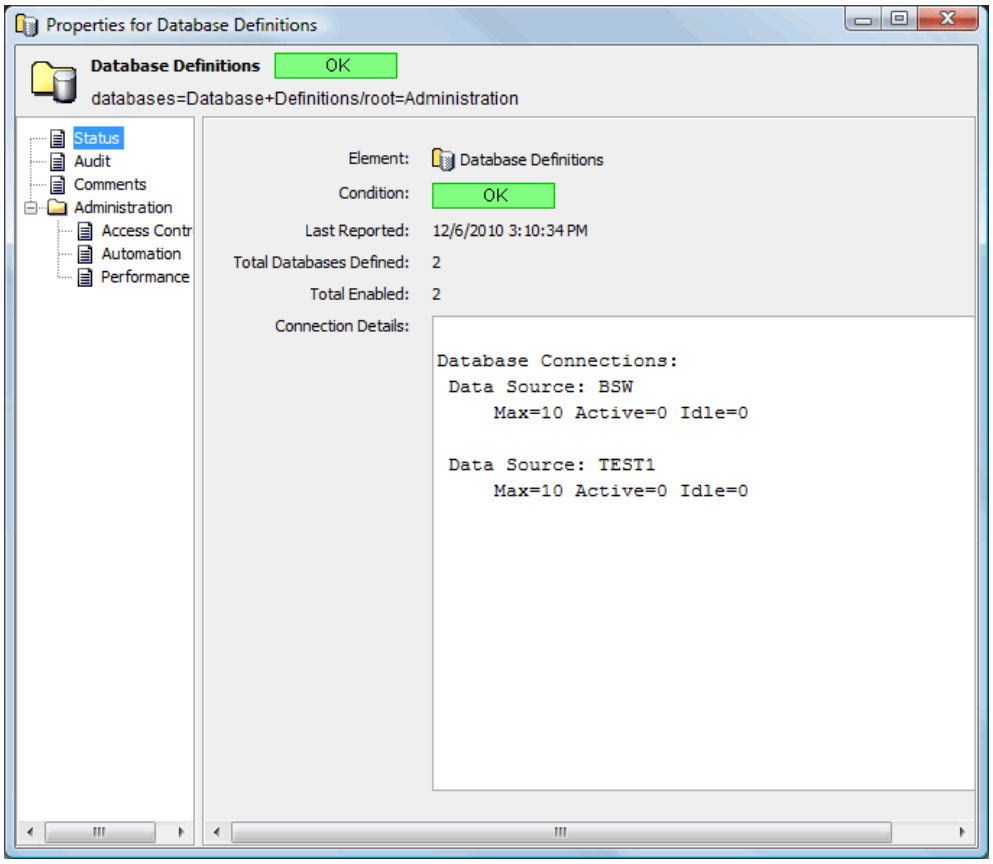
1. In the *Explorer* pane, expand the *Database Definitions* element.
2. Right-click a database definition, then select *Properties* to open the Status property page:



To view the status and statistics for all database definitions:

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

The available statistics display:



2 To disable text wrapping for easier reading, right-click inside the *Connection Details* section, then select *Word Wrap*.

7.5 Configuring the Service Warehouse

The Service Warehouse stores alarm history and comments as well as historical performance and service level data. For supported databases and specific product requirements, see the *Operations Center 5.5 Getting Started Guide*.

The Service Warehouse gathers the service level metrics required for determining whether service compliance is being met and service level health is acceptable, both of which are critical to managing service level agreements (SLAs). Management of SLAs requires the use of another Operations Center product called the Service Level Manager (SLM), which is licensed separately. For more information about SLAs and SLM, see the *Operations Center 5.5 Service Level Agreement Guide*.

Even if you are not using SLAs and SLM, you need to create and define a Service Warehouse in order to use alarm history data. For more information on alarms, see Chapter 11, "Customizing Monitored Elements and Alarms," on page 147.

The following sections cover various topics related to creating and maintaining a Service Warehouse:

- Section 7.5.1, "Sizing the Service Warehouse," on page 91
- Section 7.5.2, "Configuring the Service Warehouse," on page 95
- Section 7.5.3, "Customizing Data Collection Settings for Alarms and Performance Metrics," on page 97
- Section 7.5.4, "Enabling the Service Warehouse Backup Repository," on page 99
- Section 7.5.5, "Enabling and Disabling the Service Warehouse," on page 101
- Section 7.5.6, "Auditing Service Warehouse Events," on page 101
- Section 7.5.7, "Viewing Service Warehouse Status & Statistics," on page 102

7.5.1 Sizing the Service Warehouse

The Service Warehouse stores two basic types of data:

- Alarm history
- Service level agreement (SLA) performance and service level metric data

This data is described briefly here but for more detailed information, see the *Service Warehouse Data Dictionary*.

To assist in creating a Service Warehouse, Operations Center provides sample scripts. Prior to using these scripts, you need to determine the approximate size of the database to create. To calculate the approximate space needed by the database to accommodate the requirements of the Service Warehouse, you must closely examine your configuration and use simple arithmetic.

When estimating SLA performance and service level metric data storage requirements, it is necessary to consider:

- The number and type of expressions and data retention settings
- All historical alarm types and data retention settings
- The frequency with which elements are shared across multiple services

[The following sections cover these various considerations when planning for your Service Warehouse:]

- "Historical Alarms" on page 92
- "BSLM Performance and SLA Metric Data" on page 92
- "Basic Calculation" on page 93
- "Data Purging" on page 95

Historical Alarms

When estimating the number of alarms that will be stored, consider all types of alarms. Table 7-1 provides an overview of the five types of historical alarms data stored in the Service Warehouse.

Table 7-1 *Historical Alarm Types*

Alarm Type	Amount of Storage
Audit Alarms	Depends on specific audit settings.
Service Alarms	Varies by the number and type of adapters, and the associated number of elements generated in the element hierarchy.
Service Level Breach and Warning Alarms	Depends on the number of SLAs and objectives set per element * 2 (1 for warning and 1 for breach), the interval over which objectives are measured (hourly vs. monthly, aligned vs. rolling), and the anticipated failure frequency. It's possible that a breach alarm is issued every hour for an extended period of time when an extended outage occurs.
Outages	Depends on the number of manual outages create or imported from external data sources.
Comments	Might not impact storage requirements. Every comment on an alarm is a new comment. Note that Comments are removed when their associated alarm history is removed.

In addition, the data retention period must be factored into the storage estimates. Set data retention independently for audit, service and service level breach, and warning alarms. Alarm history is stored in the BSAAlarmData table.

Another consideration is the frequency with which elements are shared across multiple services. Table size could increase significantly depending on the frequency of shared elements within the service hierarchy. The relationship between elements and alarms is stored in BSAAlarmElements table.

BSLM Performance and SLA Metric Data

All service level metric data is managed by the Service Levels profile, which uses a data retention default of 90 days. You can customize the data retention settings to suit specific requirements. For more information on updating profile data retention settings, see "Setting Service Levels Profile Properties" in the *Operations Center 5.5 Service Level Agreement Guide*.

For service level metric data, condition changes reside in the BSAFactSeriesData table. Estimate service level metric storage requirements using a single BSLM expression in the calculation in "Basic Calculation" on page 93.

Capture performance metrics using one or more profiles, each with its own data retention setting. Set this frequency for data capture for a profile by associating a schedule. If data is stored every 5 minutes, then the calculation should include a multiplier based on this value.

Basic Calculation

These calculations are approximate. All numbers should include a small error factor to account for different types of alarm data and historical performance expressions.

The basic formula for calculating space requirements is:

(((x * 2,280) + (y * 100) * 31) * 2) = ~MB of data storage per month

Where:

- x = target number of alarms per day
- y = target number of BSLM expressions per day
- 31 = number of days in a month
- 2 = number of tablespaces (one for data and the other for indexes)

For the sake of simplicity, the number 1,000 in decimals is considered to be equal to 1K.

Some sample scenarios:

- "Example #1, Calculation Scenario" on page 93
- "Example #2, Database Report Illustration" on page 94

Example #1, Calculation Scenario

For example, assume the target is to collect each day:

- 10,000 alarms
- An additional 100,000 performance and/or SLA metric records for other BSLM profiles and expressions

For alarm data, the breakdown calculation is outlined in Table 7-2.

Table 7-2 *Alarm Data Approximations*

Approximate Bytes	Per
~ 1600	Alarm record
~7 * 40	Alarm element mapping
~0.5 * 400	Element DName storage
~200	Miscellaneous series identifiers and timeband records
~2280	Total bytes per alarm stored

Therefore, the space required for alarm data is approximately:

10,000 * 2,280 = 23 MB per day

For the additional expressions for performance and SLA data, multiply each record by 100 bytes. Therefore the example calculation is:

100,000 * 100 average bytes/expression = 10 MB per day

Using the above calculations, the combined total is 33MB per day.

To accommodate one month of data at the rate defined above, the database tablespaces can be approximately 900MB – 1GB for each tablespace.

Since there are two tablespaces for BSLM (one for data, the other for indexes), then allocate approximately 2 GB per month when creating the database.

Example #2, Database Report Illustration

As an additional illustration, the following database report was generated from an Oracle database with Historical Performance data collected for T/EC and PATROL alarms after one week of activity. In this case, there were three profiles, each with data retention set for seven days:

PATROL element with:

- 4 performance expressions
- 1 alarm expression
- 1 minute interval

Enterprise root:

- matching on '.*'
- expression on child condition count
- expression on element condition
- 10 minute interval

T/EC element with:

- 4 performance expressions
- 1 minute interval

Database Table Name	Number of Rows	Total Size (KB)	Total Size (MB)	Bytes per Row
BSAADAPTERS	1	8	.01	8192
BSAALARMDATA	41,801	53,376	52.13	1308
BSAALARMELEMENTS	220,341	7,312	7.14	34
BSADATATYPES	12	8	.01	683
BSAELEMENTS	149	24	.02	165
BSAFACTSERIESDATA *	1,575,557	142,376	139.04	93
BSASERIES	545	96	.09	180
BSATIMEBAND	39,385	2,816	2.75	73
BSAVERSIONWAREHOUSE	1	8	.01	8192

If storing Root Cause data, add an additional 1K per Root Cause entry to the BSAFACTSERIESDATA table.

Data Purging

An important factor in calculating database space requirements is the Service Warehouse data purging which is performed at scheduled intervals using the Operations Center Jobs function. Operations Center ships with a default job, the *BSW Historical Data Purge* job, which is designed to purge the Service Warehouse every morning at 3 AM (this can be modified).

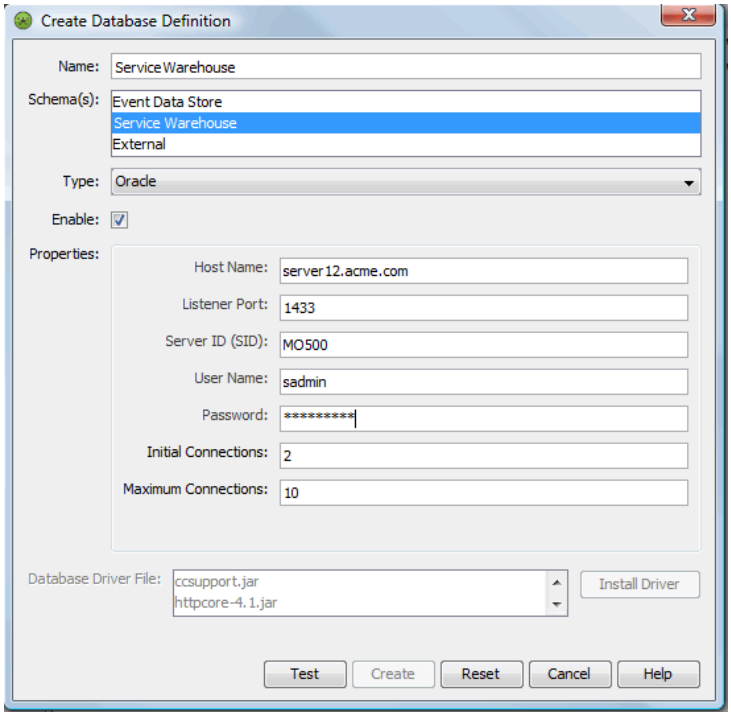
This job purges all alarm history, historical performance data, and SLA data based on the *Retain this data for x days data retention* setting defined for each profile that is capturing data. As of version 4.7 purge dates are recorded at the time the data is written to the Service Warehouse, so changes to retention settings are applied going forward.

7.5.2 Configuring the Service Warehouse

To define and configure the Service Warehouse, you'll need to create a database definition specifically for the Service Warehouse. Once this definition is created, any change to the Database Type requires a restart to the Operations Center server.

To configure and enable the Service Warehouse:

1. In the Operations Center console *Explorer* pane, expand *Administration* > *Database Definitions*.
2. Right-click *Database Definitions*, then select *Create Database Definitions* to open the Create Database Definition dialog.



3. Enter a name for the definition in the *Name* field.

   It is best practice to use a name without spaces or special characters.
4. In the Schema(s) section, make sure that *Service Warehouse* is selected.
5. Click the *Type* drop-down list, then select a database type.

If selecting *(SQL Server (Domain Authentication)*, there are required configurations for Windows servers. For more information, see Section 7.1, "Configuring Windows Servers for Single Sign On (SSO)," on page 70.

NOTE: Once the Service Warehouse definition is created, any change to the Database Type requires a restart to the Operations Center server.

6 Select the *Enable* check box to activate the database definition.

7 Specify the Database Properties required to establish the database connection (the required properties vary depending on the database):

Database Properties	Description
Hostname	The name of the database server.
Listener Port	The port on which the database listens for communications.
Server ID (SID)	The name of the database (used when defining an Oracle database connection).
Database	The name of the database (used when defining a Microsoft SQL Server, Sybase, or DB2 database connection).
Domain	The domain to use for domain authentication with single sign on (used when defining a Microsoft SQL Server database with Domain Authentication).
User Name	The name of the user account. When multiple Operations Center schemas exist within the same database, a unique ID must be created for each database schema. If using Microsoft SQL Server with Domain Authentication, specify the username of the Windows user account, or leave blank (on Windows servers) to attempt to use the credentials of the currently active Windows user account.
Password	The password for the user account. If using Microsoft SQL Server with Domain Authentication, specify the password of the Windows user account, or leave blank (on Windows servers) to attempt to use the credentials of the currently active Windows user account.
Initial Connections	The number of connections established upon initial connection.
Maximum Connections	The maximum number of connections allowed.
Additional Properties	(Optional) Additional JDBC URL properties for Microsoft SQL Server and Microsoft SQL Server (Domain Authentication) databases. Prefix each parameter entry with a semi-colon. For example, `;parameter1=value1;parameter2=value2`
Database Class, URL, and Driver File	Name and location of the IBM DB2 driver. If using an IBM DB2 database, it is necessary to upload the JDBC driver file supplied by IBM.

When setting the number of database connections allowed, calculate the number of required database connections by analyzing your user base and other Operations Center settings. Identify the following:

- The number of database connections required in part depends on how many users access reports.

- The number of database connections used for storing alarm history has a one-to-one correspondence to the number of alarm threads configured for the Service Warehouse.
- The number of database connections used for storing Performance Series data has a one-to-one correspondence to the number of performance threads configured for the Service Warehouse.

Then, use the following calculation to estimate the number connections needed:

```
(# users * 0.5) + (# of Alarm Threads * 1) + (# Series Threads * 1) +10
```

The Service Warehouse uses a connection pool, therefore each connection is returned to the connection pool when the query is complete. If the number of used connections is always at the maximum, then increase the number of connections by 20 percent.

8 Click the *Install Driver* button to browse, then select a driver file.

9 Click the *Test* button to test and verify that the database settings are valid, prior to creating the definition.

If the database connection is valid, the *Create* button activates.

IMPORTANT: If you are using Operations Center in a clustered environment, one Operations Center server must be designated as the primary server writing to the Service Warehouse. This is indicated in Configuration Manager on the *Database* pane on the *Components* tab. On one configuration, change the *Primary Warehouse Writer* option to True. By default all others are False.

For information on clustered environments, see "Implementing a High Availability Solution" in the *Operations Center 5.5 Server Installation Guide*.

7.5.3 Customizing Data Collection Settings for Alarms and Performance Metrics

The Service Warehouse is preconfigured with default settings that allow Operations Center software to capture and record alarm and performance data. These settings can be adjusted depending on the various needs of your system.

System performance can vary based on the combined settings specified for Performance Threads, Transactions Per Commit, and Purge Block Size. A higher transaction limit provides better performance. However, it can increase database recovery time, and the potential of locking out other threads trying to access the same tables. If set too high, the database can also run out of database cursors.

Table 7-3 shows the setting values that are likely to produce the best performance levels for most systems.

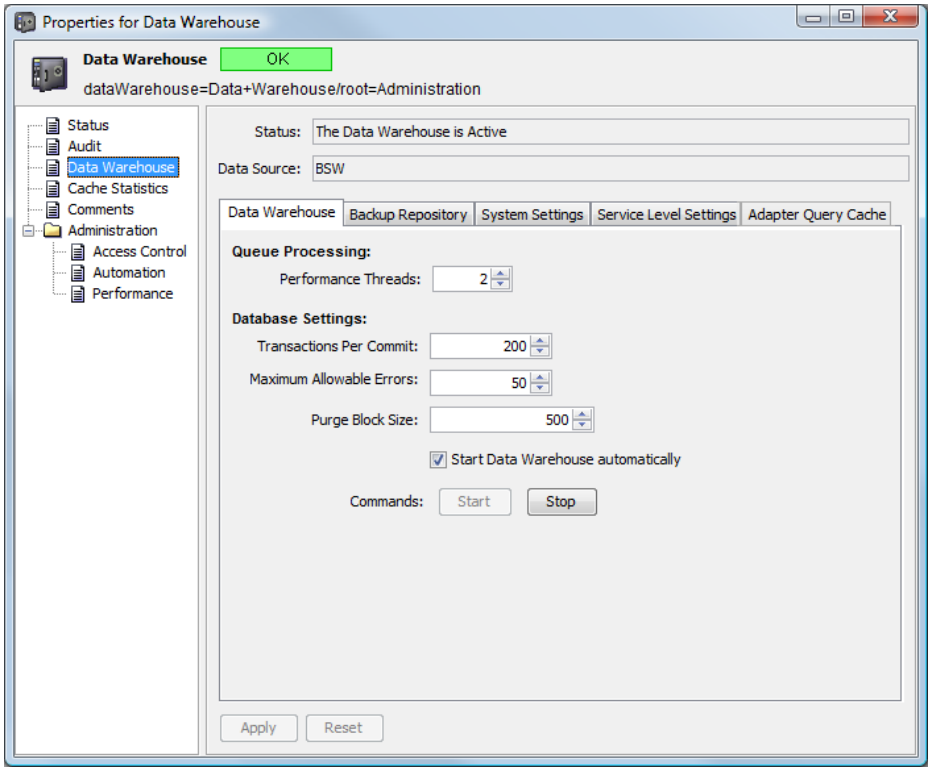
Table 7-3 *Recommended Settings for Most Systems*

Setting	Sybase	Oracle	MSSQL
Performance Threads	4	4	4
Transactions Per Commit	50	50	50
Purge Block Size	1,000	2,000	1,000

The Service Warehouse collects alarm and performance data using an element called *Data Warehouse* which appears in the hierarchy in the Operations Center console under *Enterprise* > *Administration*. For information about collecting data for service level agreements, see the *Operations Center 5.5 Service Level Agreement Guide*.

To edit or configure Data Collection settings:

1. In the *Explorer* pane, expand the root *Administration* element.
2. Right-click *Data Warehouse*, and select *Properties*.
3. Select *Data Warehouse* in the left pane.



4. (*Data Warehouse* tab) Configure the following options for the capturing, recording, and maintenance of alarm and performance data:
   - **Performance Threads:** The number of threads to handle historical performance data.

     The queue for Performance data can have a maximum of eight threads with each thread using one or two database connections.
   - **Transactions Per Commit:** The number of objects per commitment.

     A higher transaction limit provides better performance. However, it can increase database recovery time, and there is the potential of locking out other threads trying to access the same tables. If set too high, the database can also run out of database cursors. It is suggested to use a setting of 50.

- **Maximum Allowable Errors:** The maximum allowable errors before the Service Warehouse shuts down.

  NOTE: When the threshold is exceeded (i.e. when the database runs out of disk space or the database is unavailable for any reason), the Service Warehouse automatically initiates the Backup Repository, if enabled. If the Backup Repository is not enabled, data processing is shutdown entirely.

  When the database becomes available again, it is necessary to disable and then reenable the Database Definition (created in Section 7.5.2, "Configuring the Service Warehouse," on page 95) which alerts the Service Warehouse to switch from backup mode and return to using the database.

- **Purge Block Size:** The number of objects to clear from the database with each purge transaction.
- Select the *Start Data Warehouse Automatically* check box to start the Data Warehouse when the Operations Center server is started.

5 (*Adapter Query Cache* tab) Configure cache settings for responses from integration systems. The data returned from adapter queries can be cached to help speed response times.

- **Adapter Query Cache Size:** The number of adapter responses to cache.
- **Adapter Query Cache TTL:** The retention time (in seconds) to hold query responses in the cache.

7.5.4 Enabling the Service Warehouse Backup Repository

The Service Warehouse has a backup repository system which uses a file system that stores excess data until queues return to normal and data in the backup files is processed.

When enabled, Operations Center uses the in-memory queue and writes data to a backup repository in the event that:

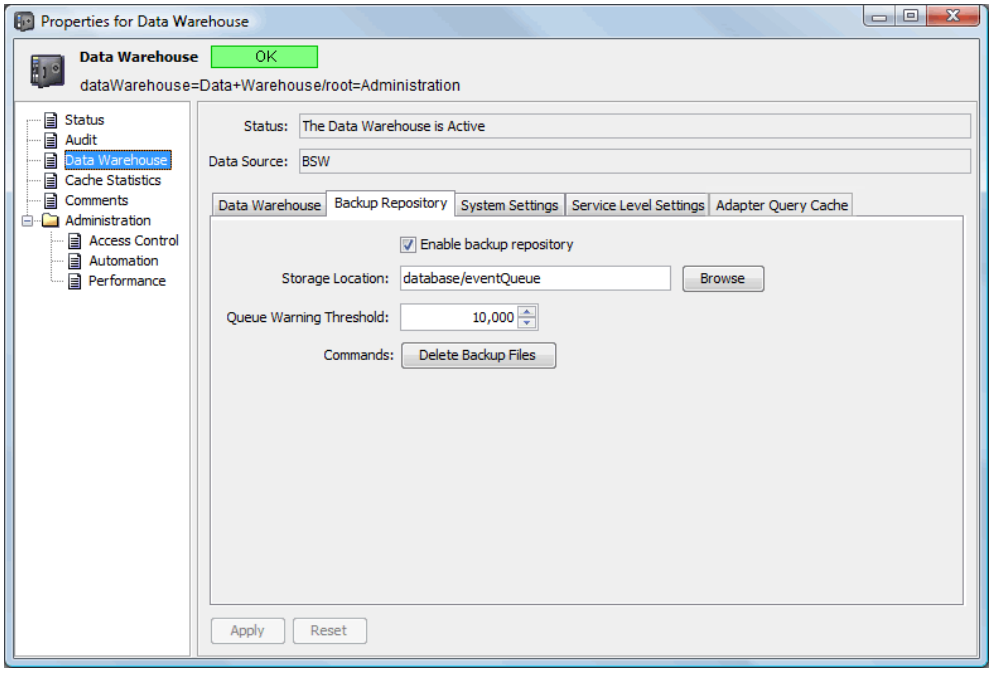
- The Service Warehouse database connection fails.
- The database temporarily shuts down.
- The in-memory queue for writing historical data to the database exceeds the *Queue Maximum* setting, when *Warehouse Write Mode* is set to `MEMORY-QUEUED` in the Configuration Manager. For more information on the *Queue Maximum* database setting, see Step 5 on page 100.

  For more information on Configuration Manger settings, see Section 2.2.11, "Database Pane," on page 35.
- The *Maximum Allowable Errors* setting is reached and exceeded (for example, when the database runs out of disk space or the database is unavailable for any reason) and the Service Warehouse shuts down. For more information on the *Maximum Allowable Errors* database setting, see Step 4 on page 98.

  In this case, it is necessary to re-enable the database definition after the problem is solved and the database is available. For more information see Section 7.3.5, "Disabling and Deleting a Definition," on page 89.

To enable or configure settings for the backup repository:

1. In the *Explorer* pane, right-click the *Data Warehouse* root element, then select *Properties* to open the Status property page.
2. In the left pane, click *Data Warehouse* to open the Data Warehouse property page, then select the *Backup Repository* tab:



3. Select the *Enable Backup Repository* check box to activate the back repository process.

   If implementing Service Level Management (SLM), enabling the backup repository is required. For more information about SLM, see the *Operations Center 5.5 Service Level Agreement Guide*.

4. (Optional) To specify the location of the backup repository, enter the directory for the backup repository files in the *Storage Location* field, or click *Browse* to navigate, then select a directory.

   The backup repository is configured and enabled by default to store data in the `/OperationsCenter_install_path/database/eventQueue` directory.

5. (Optional) Specify the following settings when *Warehouse Write Mode* is set to `MEMORY-QUEUED` in the Configuration Manager:

   Maximum Disk Space: The maximum amount of disk space used for backup repository files.

   Maximum File Size: The maximum size of a generated backup file before an additional backup file is created.

   Queue Maximum: The maximum size for the database queue, above which additional collected data is written to backup repository files.

   Queue Minimum: The minimum size for the database queue, below which any data stored in backup repository files is reissued to the queue.

6. (Optional) To process existing backup repository files when the Service Warehouse starts, select the *Process Backup Files Automatically* check box.
7. (Optional) To delete all existing backup repository files, click the *Delete Backup Files* button.
8. Click *Apply* to save the changes.

7.5.5 Enabling and Disabling the Service Warehouse

If the database definition for the Service Warehouse is disabled, data is collected, but it is stored in the backup repository only. During this time, reports and views cannot show the most recent data until the Service Warehouse database is back online and data in the backup repository is written to the database.

WARNING: If the maximum number of allowable errors is reached (i.e. when the database runs out of disk space or the database is unavailable for any reason), the Service Warehouse shuts down. After troubleshooting/resolving the issue and the database is available, you must disable and then reenable the Database Definition. By reenabling the database definition, the Service Warehouse is alerted that the database is available again.

For more information on the *Maximum Allowable Errors* database setting, see Step 4 on page 98.

To manually enable or disable the Service Warehouse, right-click the Service Warehouse element, then select either *Enable Database Definition* or *Disable Database Definition*.

7.5.6 Auditing Service Warehouse Events

Updates to the Service Warehouse can be audited. The following events can be audited:

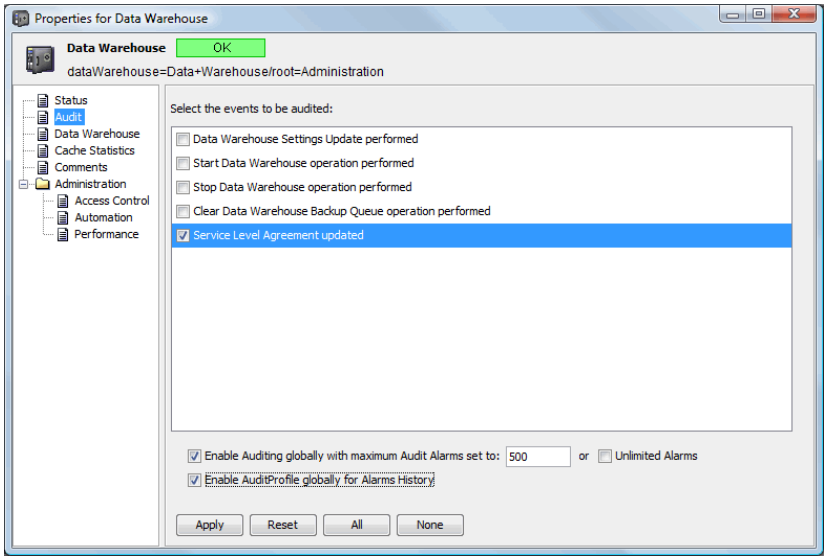
- Service Warehouse settings updated
- Start or stop the data warehouse engine
- Clear the Service Warehouse backup queue

Audit events display in the *Audit Event* channel in the *Alarms* view on the *Data Warehouse* element. For information about audited events, see the *Operations Center 5.5 Security Management Guide*.

Auditing is set in the *Explorer* pane, under *Administration*. Right-click *Data Warehouse*, then select *Properties*. On the *Audit* tab, select the events to audit. Audit events display in the *Alarms* view. For more information on alarms, see Chapter 11, "Customizing Monitored Elements and Alarms," on page 147.

To set auditing:

1. In the *Explorer* pane, expand *Administration*.
2. Right-click *Data Warehouse*, then select *Properties* to open the Status property page.
3. In the left pane, click *Audit* to update the *Audit* property page:



4. Select the events to audit.
5. Click the *Apply* button.

7.5.7 Viewing Service Warehouse Status & Statistics

Cache statistics are available for the data warehouse for the Service Warehouse.

To view the status and statistics for the Data Warehouse:

1. In the *Explorer* pane, right-click the *Date Warehouse* root element, then select *Properties*.

   The Status property page opens and displays information about Repository Status, Queue Size, and Repository Details.

   TIP: To disable text wrapping for easier reading, right-click the *Repository Details* pane, then select *Word Wrap*.

   ---

2. In the left pane, click *Data Warehouse* to open the Data Warehouse property page.
3. To view warehouse settings and service level settings click the *Data Warehouse*, *Backup Repository*, *System Settings*, or *Service Level Settings* tabs.
4. To view caching statistics for primary keys, alarm metadata, profiles, and time series information, click the *Cache Statistics* tab.

   TIP: To disable text wrapping, right-click the *Cache Statistics* pane, then select *Word Wrap*.

7.6 About Operations Center Embedded Databases

Various Operations Center products use embedded databases, mostly to store configuration data. They are:

- Section 7.6.1, “The Dashboard,” on page 103
- Section 7.6.2, “SQL Views,” on page 103

7.6.1 The Dashboard

By default, the dashboard is installed with its own Hypersonic SQL database embedded in the software to store configuration and portlet information. However, the dashboard must be configured to use an external database. For more information about configuring the dashboard, see the *Operations Center 5.5 Dashboard Guide*.

7.6.2 SQL Views

SQL Views uses an Apache Derby database that is embedded and installed with Operations Center to make data available for use with SQL Views. Apache Derby is a relational database implemented entirely in Java. SQL Views does not support the use of any other type of database.

No database definition is required for the SQL Views database. *Operations Center* appears as an element in the Operations Center console in the hierarchy under *Enterprise* > *Administration* > *SQL Views*.

For more information about SQL Views, see the *Operations Center 5.5 SQL Views Guide*.

8 Managing Configurations

A configuration for the Operations Center server is defined as the hierarchy of elements in the Operations Center server as well as relationships between the objects, data collection settings, and other data and parameter settings. Configurations are XML files and are stored in Configuration Storage.

Configurations can be created and modified, and copied via import and export functionality.

When working with configurations, you might be prompted to enter a user ID and password. You must enter the ID and password for the default admin account for Operations Center. For more information about user IDs and passwords, see the *Operations Center 5.5 Security Management Guide*.

- Section 8.1, "Using the Configuration Explorer," on page 105
- Section 8.2, "Setting the Configuration Default," on page 108
- Section 8.3, "Creating Configurations," on page 108
- Section 8.4, "Modifying Configurations," on page 110
- Section 8.5, "Backing Up, Copying, and Restoring Configurations," on page 112

8.1 Using the Configuration Explorer

NOTE: Configurations are managed using the Configuration Explorer. It is necessary to stop the Operations Center server before performing actions in the Configuration Explorer.

- Section 8.1.1, "Accessing Configuration Explorer," on page 105
- Section 8.1.2, "Understanding the Menu Options," on page 106

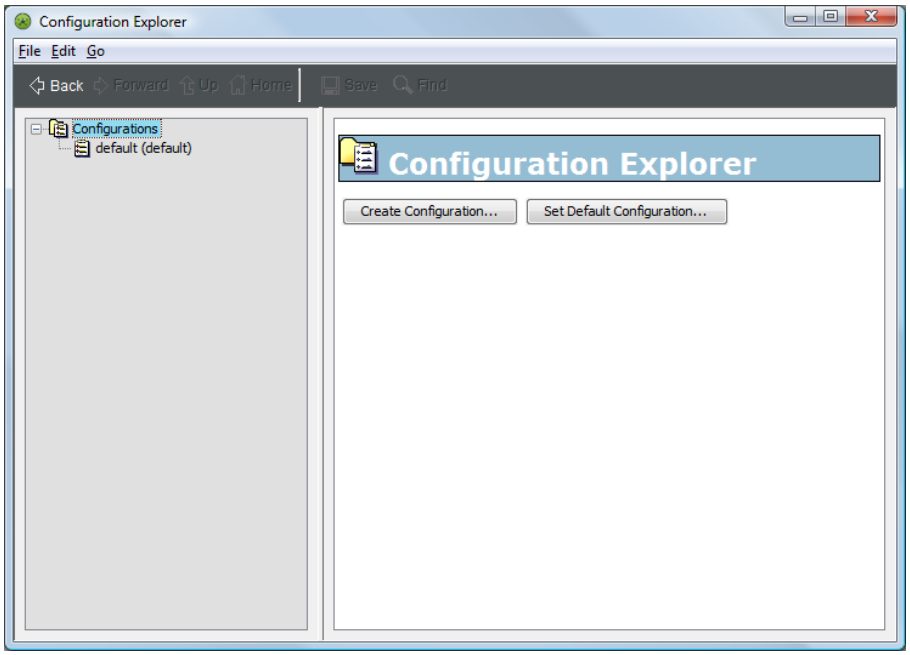
8.1.1 Accessing Configuration Explorer

To access the Configuration Explorer:

1 To access the Configuration Explorer, do one of the following:

- In Configuration Manager, click the *Explore* button for the *Configuration Storage* option on the *Server* pane on the *Components* tab.
- Enter `moscfg` at command line.

Upon opening the Configuration Explorer, you are prompted to open (optionally) the default configuration.



Each configuration is listed in the hierarchy under *Configurations*. The default is automatically created when Operations Center is installed. Under each configuration is the hierarchy of elements from the Operations Center server.

2 To view the actions that are available from the menu bar at the top, right-click *Configurations* or an element under *Configurations*.

IMPORTANT: The Operations Center server should be stopped prior to performing any actions in the Configuration Explorer.

8.1.2 Understanding the Menu Options

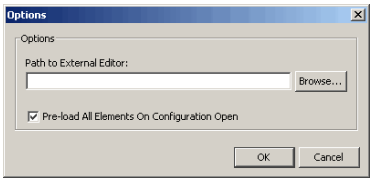
- "Main Menu Options" on page 106
- "Right-Click Menu Options" on page 108

Main Menu Options

Table 8-1 describes the menu options available in the Configuration Explorer.

Table 8-1 *Configuration Explorer Main Menu Options*

Menu	Option	Explanation
File	*New Configuration*	Creates a configuration.
	Save	Saves a modified configuration.
	Save As	Saves the selected configuration using a different name.

Menu	Option	Explanation
	Copy To	Copies the selected configuration to a different configuration.
	Revert	Opens the last saved version of a configuration.
	Import	Imports a configuration. Select an XML file, then click the *Open* button.
	Export	Exports a configuration to an XML file. Provide a name for the export file. If you receive an error during export, click *Edit* > *Options* and verify that the check box is selected.
	Exit	Closes the Configuration Explorer window.
Edit	*Find* or *Find Next*	Locate elements based on element name/class or property name/value.
	Options	Enter the path to an external editor. Mark the check box to load all elements in memory when the configuration is opened in the Configuration Explorer. If unmarked, elements are loaded on demand. 
Go	Forward	Navigate forward in the configuration views.
	Backward	Navigate back in the configuration views.
	Up	Navigate upwards in the configuration elements tree.
	Home	Return to the default configuration.

Right-Click Menu Options

The options listed in Table 8-2 are available when you right-click a configuration in the left pane of the Configuration Explorer.

Table 8-2 *Configuration Explorer Right-Click Menu Options*

Option	Description
Change	Opens the Change Configuration Storage Definition dialog box so you can modify the Configuration Storage data source parameters.
Reinitialize	Removes all data from the configuration and reinitializes it. Displays a confirmation screen for removing all data.
Purge	Purges the data store based on the Retention Days setting. Only applies to Version Tracking configurations. For more information about Version Tracking, see the *Operations Center 5.5 Version Tracking Guide*.
Close	Closes the configuration.
Set Default Configuration	Sets the configuration as the default configuration.
Remove	Deletes the configuration.
Remove Children	Removes all child elements of the configuration.
Rename	Enter a new name for the element associated with the configuration.

8.2 Setting the Configuration Default

Multiple configurations can be associated with a Operations Center server. The server can only actively use on configuration at a time. The default is the configuration currently in use.

To set the default, right-click a configuration, then select *Set Default Configuration*.

8.3 Creating Configurations

To create a configuration, you need to specify the following:

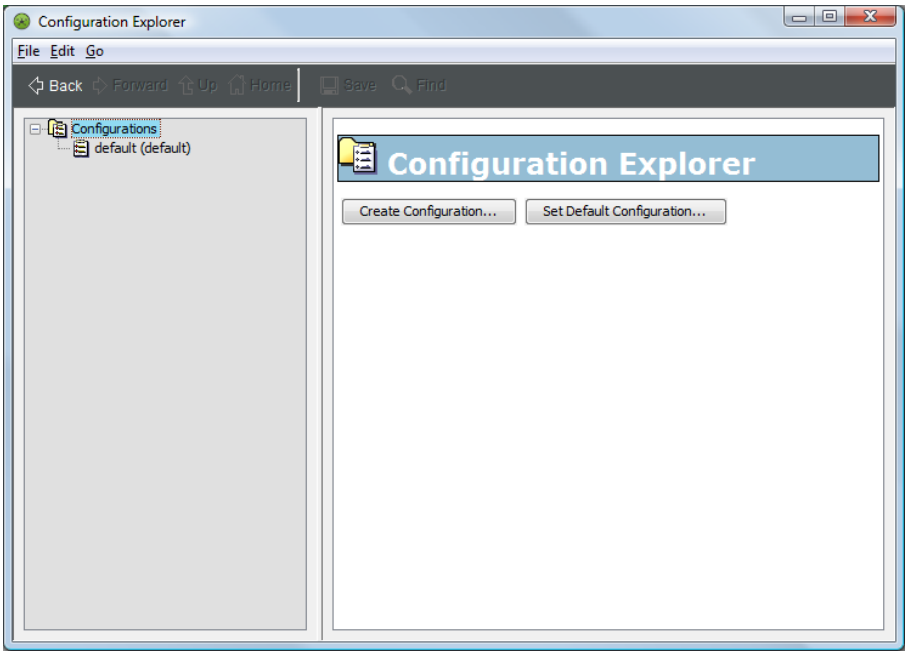
- Name of the configuration.
- Configuration storage type to indicate the type of database being used for Configuring the Database for Configuration Storage (page 70) where the configuration will be stored. The default is an Object ODB database which is embedded and installed with Operations Center.
- Parameters for Configuration Storage, which varies depending on the storage type. If you are using the embedded Object ODB database, specify only the file name of that database.

Obtain the above information before continuing.

To create a configuration:

1 In the Configuration Explorer, do one of the following:

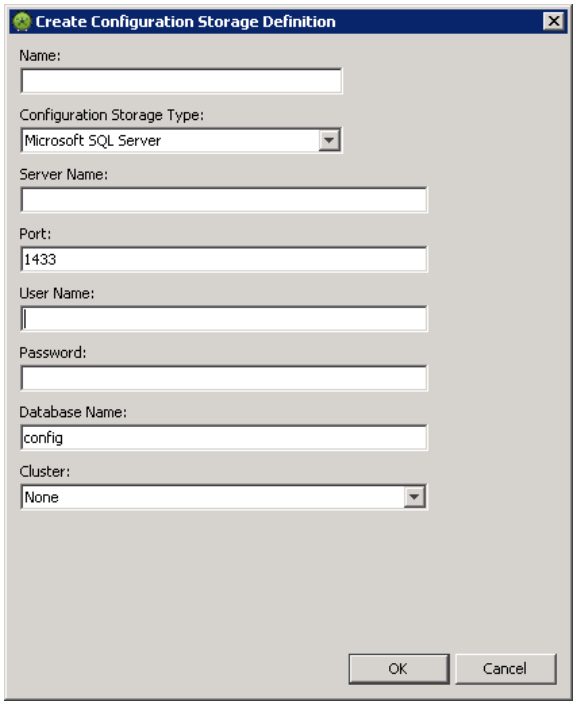
- In the left pane, click *Configurations* to update the right pane:



  Click *Create Configuration*.

- Click *File*, then select *New Configuration*.

The Create Configuration Storage Definition dialog box opens.



2 Enter the name for the new configuration in the *Name* field.

3 Select a storage type from the *Configuration Storage Type* list.

The remaining fields update and vary by storage type (Microsoft SQL, Oracle, and so on).

If configuring a cluster environment, see "Configuring Operations Center for Clustering" in *Operations Center 5.5 Server Installation Guide*.

4 Click *OK*.

8.4 Modifying Configurations

After creating a configuration, you can modify the defaults. You can also modify the configuration at any time.

- Section 8.4.1, "Changing Configuration Storage Type," on page 110
- Section 8.4.2, "Adding an Element or Property," on page 110
- Section 8.4.3, "Renaming Elements," on page 111
- Section 8.4.4, "Removing Children," on page 111
- Section 8.4.5, "Reinitializing and Deleting," on page 111

8.4.1 Changing Configuration Storage Type

After creating a configuration, you can change the type of Configuration Storage and its parameters.

To change the configuration storage type, right-click the configuration, then select *Change*.

The parameters are the same as when the configuration was created.

8.4.2 Adding an Element or Property

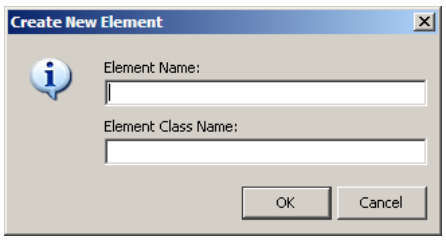
It is possible to add an element or property to a configuration:

- "Adding an Element to an Existing Configuration" on page 110
- "Adding a Property to an Existing Configuration" on page 111

For more information about elements and properties, see *Viewing Element Properties* in the *Operations Center 5.5 User Guide*.

Adding an Element to an Existing Configuration

1 In the right pane of the Configuration Explorer, click the *Add Element* button to open the Create New Element dialog box:

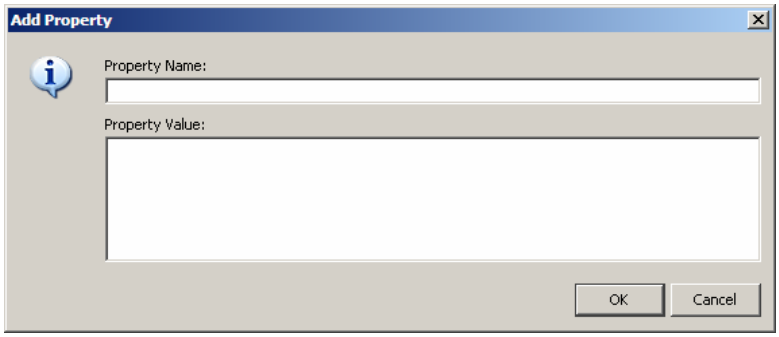


2 Fill in the *Element Name* and *Element Class Name* fields.

3 Click *OK*.

Adding a Property to an Existing Configuration

1 In the right pane of the Configuration Explorer, click the icon beside Properties to open the *Add Property* dialog box:



2 Fill in the *Property Name* and *Property Value* fields.

3 Click *OK*.

8.4.3 Renaming Elements

1 To rename an element, right-click the element, then select *Rename*.

2 Enter the new name, then click *OK*.

For more information on elements, see *Monitoring Elements and Element State* in the *Operations Center 5.5 User Guide*, and Chapter 11, "Customizing Monitored Elements and Alarms," on page 147.

8.4.4 Removing Children

To remove the children of an element, right-click the element, then select *Remove Children*.

For more information on elements and their relationships to each other, see the *Operations Center 5.5 User Guide* and the *Operations Center 5.5 Service Modeling Guide*.

8.4.5 Reinitializing and Deleting

IMPORTANT: When a configuration is reinitialized, all the data is removed from it.

To reinitialize, right-click a configuration in the Configuration Explorer, then select *Reinitialize*.

To delete a configuration, right-click the configuration in the Configuration Explorer, then select *Remove*.

8.5 Backing Up, Copying, and Restoring Configurations

Configurations are copied for two main reasons: to have a backup copy to use for a system restore if necessary and to copy a configuration from one machine to another.

An Operations Center configuration might need to be copied from one machine (source) to another (target) machine for other reasons including:

- Promoting a test server configuration to a production server
- Two identical Operations Center servers are configured for fault tolerance (high availability) purposes
- A backup copy of customizations to the Operations Center configuration needs to be created without retaining the entire directory structure
- After upgrading Operations Center, copy the configuration settings from the default database to a new external database

There are multiple ways to copy a full or partial configuration from one Operations Center server to another:

- Backup or copy the entire server configuration by copying the Configuration Storage database file. See Section 8.5.1, “Using the Configuration Storage Database File,” on page 112.
- Backup or copy a configuration using the Configuration Explorer. See Section 8.5.2, “Using the Configuration Explorer,” on page 113.
- Backup or copy a configuration on UNIX or Linux systems from the command line. Section 8.5.3, “Using the Exportcfg and Importcfg Utilities,” on page 114.
- Import or export a full or partial configuration using console Import and Export features while the server is running. See Section 8.5.4, “Using Import/Export from the Operations Center Console,” on page 116.
- Import or export a full or partial configuration using NOC script.Section 8.5.5, “Using NOC Script to Import and Export Configuration XML Files,” on page 119.
- Export the complete configuration into Object database format

When importing and exporting files, verify that the Operations Center installation directory name and location are consistent on the source server and the target server.

- Section 8.5.1, “Using the Configuration Storage Database File,” on page 112
- Section 8.5.2, “Using the Configuration Explorer,” on page 113
- Section 8.5.3, “Using the Exportcfg and Importcfg Utilities,” on page 114
- Section 8.5.4, “Using Import/Export from the Operations Center Console,” on page 116
- Section 8.5.5, “Using NOC Script to Import and Export Configuration XML Files,” on page 119
- Section 8.5.6, “Using NOC Script to Backup the Configuration to an ODB Database File,” on page 124

8.5.1 Using the Configuration Storage Database File

Copy entire configurations from an Operations Center server by copying one file, when

- The configuration is stored in a Configuration Storage data store located in the embedded Object ODB database (which is the default) versus an external database.
- Both Operations Center servers are stopped.

To copy a Operations Center configuration when both servers are not running:

1. If the target Operations Center server has Operations Center software installed, delete the `/OperationsCenter_install_path/configstore/.$default.odb$` file.
2. Shut down both servers.
3. Copy the `/OperationsCenter_install_path/configstore/default.odb` file that contains the configuration to the `/OperationsCenter_install_path/configstore` directory on the target server.

   Do not copy any other files.
4. Restart the target Operations Center server.

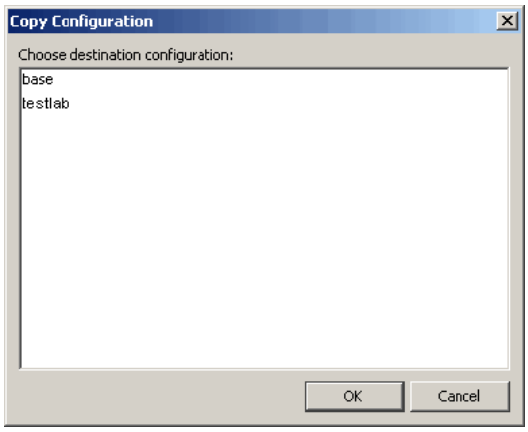
8.5.2 Using the Configuration Explorer

Use the Configuration Explorer when both Operations Center servers are stopped to either copy the open configuration to another configuration or to import and export full configurations:

- "Copying a Configuration from One Server to Another" on page 113
- "Exporting a Configuration" on page 114
- "Importing a Configuration" on page 114

Copying a Configuration from One Server to Another

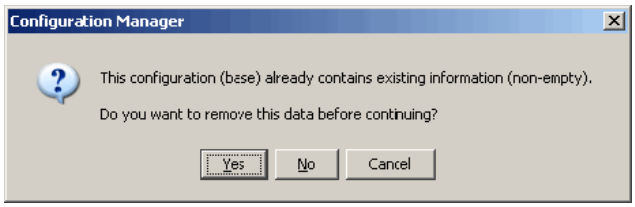
1. In the left pane of the Configuration Explorer, select a configuration to copy.
2. Click *Edit* > *Copy* to open the Copy Configuration dialog box:



3. In the Copy Configuration dialog box, select a destination configuration.
4. Click *OK*.

   The current configuration is copied the other configuration.

5 If the target configuration contains data and the following dialog box displays, overwrite the existing data by clicking *Yes*:



Exporting a Configuration

1 Select the configuration you want to export.

2 Click *File* > *Export*.

3 Select a location and file name, then click *OK*.

If you receive an error, verify that the option to preload all elements on configuration open in selected in the dialog box accessed under *Edit* > *Options*.

Importing a Configuration

1 Click *File* > *Import*.

2 Select a configuration file.

3 Click *OK*.

8.5.3 Using the Exportcfg and Importcfg Utilities

Use the `exportcfg` and `importcfg` utilities to copy a Operations Center configuration on UNIX or Linux systems (Linux, Solaris), which requires access to an X display. The utilities do not display a user interface, but use the X display to initialize a hidden client that communicates with the server.

Use the `exportcfg` and `importcfg` when the Operations Center servers are running.

To copy an Operations Center configuration using the `exportcfg` and `importcfg` utilities:

1 At a command prompt on the server that contains the configuration to copy, enter the `exportcfg` command:

- `exportcfg` *`dname filename`*

  For example,

  ```
  exportcfg "" allData.config.xml
  exportcfg root=Locations Locations.config.xml
  ```

- To export for use in an on-line backup:

  `exportcfg` *`filename hostname port userid password dname`* `[`*`exportRelationships`*`] [`*`exportRelatedElements`*`] [`*`propertyFilter`* `[`*`excludeFilter`*`]`

  For example,

  ```
  exportcfg all.config.xml localhost 80 admin formula root=Organizations
  ```

- To attempt to connect to the named configuration for use in off-line backup while the server is down:

```
exportcfg -cold configurationName userid password dname
[exportRelationships] [exportRelatedElements] [propertyFilter]
[excludeFilter
```

The `-cold` command can only be run locally. If using an object-oriented database, you must stop the server before using the `-cold` command.

For example,

```
exportcfg -cold all.config.xml MyDB admin formula root=Organizations
```

The following table explains the replaceable parameters used with the `exportcfg` command:

Replace…	With…
filename	The name of the configuration file created on the Operations Center server to which the configuration is being copied.
hostname	The Operations Center server name.
port	The port number for the Operations Center server.
configurationName	Name given to the configuration specified in Configuration Manager. Default: `default`.
userid	User ID with Administrator rights used to access the Operations Center server. Providing the User ID prevents the server from requesting it.
password	Password for the User ID with Administrator rights used to access the Operations Center server.
dname	The DName of the element to export. Use “” for the Enterprise root.
exportRelationships	Exports all relationship with the configuration, expressed as True or False.
exportRelatedElements	Exports elements the relationships refer to. If elements are defined on the server to which the configuration is being copied, the `importcfg` command overwrites them.
propertyFilter	Named properties that are to be exported.
excludeFilter	As an inverse of the above option, use this option to specify named properties that are not to be copied, based on a regular expression.

2 Enter `importcfg` at a command prompt on the server on which the configuration is to be installed using the following syntax:

```
importcfg filename removeDeleted
```

The following table explains the replaceable parameters used with the `importcfg` command:

Replace…	With…
filename	The name of the configuration file to be imported.
removeDeleted	`true`: existing elements and relationships not referenced by the configuration file are removed. `false`: all existing elements and relationships remain on the server. To learn more about this functionality, see Step 3 on page 118 in Importing Configuration Files.

For example:

```
importcfg allData.config.xml true
```

```
importcfg Locations.config.xml false
```

3. Enter the Web server hostname at the command prompt.
4. Enter the Web server port at the command prompt.
5. If you did not provide your User ID and password as part of the `exportcfg` command, enter the User ID and password at their respective command prompts.

   The configuration is copied.

8.5.4 Using Import/Export from the Operations Center Console

Configuration imports and exports of full or partial configurations can be done via the Operations Center console. The advantage of the Configuration Import/Export feature is that the Operations Center servers need not be stopped.

The console Import/Export feature allows you to copy a specified hierarchy of elements from one server and import it into another server including the:

- Element relationships
- Related elements
- Element properties
- SCM definitions
- Layout SVG drawings
- SLA definitions including SLA calendars (applies to Service Model elements only)

The console Import/Export feature was designed to copy the *Services* > *Service Models* hierarchy, but can also be used an some *Administration* branches such as:

- *Adapters*
- *Metamodel*
- *Graphics*
- *Security*
- *Server* > *Algorithms*
- *Server* > *Operations Definitions*
- *Time Management* > *Calendars* (Does not include Blackout Calendars which are exported along with Element Properties.)

However, the console Import/Export feature is NOT used for Database Definitions, Data Warehouse, Jobs, BDI adapter definitions, Automation, Scripts, Schedules, Time Categories, Service Level Agreements.

The Operations Center console has import and export functionality available:

- "Exporting a Configuration File" on page 117
- "Importing a Configuration File" on page 118

Exporting a Configuration File

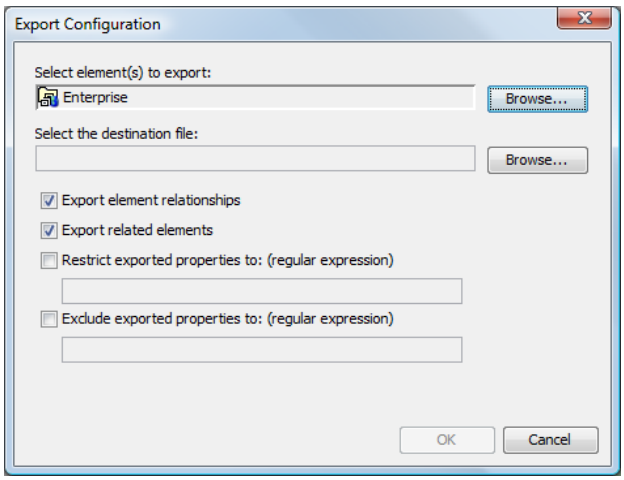
If elements are already defined on the target server and you copy only the relationships, the relationships remain intact. If you copy the related elements, then the elements and their properties overwrite those on the target server.

As an example of exporting relationships with a server configuration, assume that you want to export a view with ACLs intact, but you do not want to copy the user credentials to the target system. The relationship is between the ACL and the user. If you do not export related elements, then the user's password is left intact on the target system.

If the target server doesn't have the related elements, they can be created. This can occur in the case of a relationship between an element in the *Service Models* hierarchy and an element in the *Elements* hierarchy. In this case, to construct the relationship, the target element must be defined in Configuration Storage. It does not, however, affect any potential algorithm settings, or other property data related to the target element.

To export a configuration file:

1. In the *Explorer* pane, expand *Administration*.
2. Right-click *Server*, select *Configuration* > *Export* to open the Export Configuration dialog box:



3. For *Select Element(s) to Export*, click *Browse* and navigate to the elements to export, then click *OK* to display the selected elements.

   Select the portion of the tree to export. All elements under the selected element are exported as well as all the relationships. To export only the elements and no relationships (just the *Elements* hierarchy), select *Elements* in the hierarchy under *Enterprise*. To export the entire hierarchy, select *Enterprise*.
4. For *Select the Destination File*, click *Browse* and navigate to the location where you want to export the element, then click *Open* to display the selected location in this field.

5 Specify the items to export: with the following settings:

Export Element Relationships: Select this option to export only the relationships and not the elements themselves, if the elements are defined on the target server. Property data is not exported.

Export Related Elements: Select this option to export the elements referred to in the relationships and the associated properties. If imported, the elements and properties on the target server are overwritten.

Restrict Exported Properties To: Use this option to specify properties to copy, based on a regular expression. For example, specify `SVG_Drawing`, and no ACLs or other property data other than the `SVG_Drawing` property are exported. Assume that you export drawings associated with the elements in a portion of the hierarchy. Use the wildcard "*" to identify any text. For example, use `SVG.*` to export the same properties as `SVG_Drawing.` Use the "`|`" (pipe) character to separate a list of properties. For example, `SVG.*|acl.*` exports SVG Drawings as well as ACLs.

Exclude Exported Properties To (Regular Expression): As an inverse of the above option, use this option to specify properties that are not to be copied, based on a regular expression.

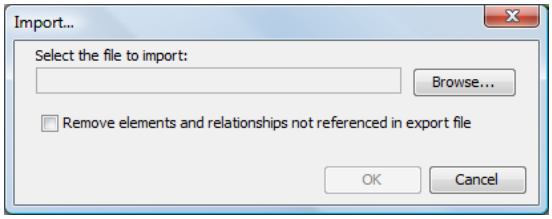
6 Click *OK* to export the configuration.

Importing a Configuration File

Before importing a configuration file, be sure the file meets the requirements for well-formed imported XML code containing necessary DTD headers, and the basic XML tags and commands for elements and views.

To import a configuration file:

1 In the Operations Center console *Explorer* pane, right-click the *Server* element, then select *Configuration* > *Import* to display the Import dialog box:



2 In the *Select the File to Import* field, click *Browse* and navigate to the location to which you want to import the configuration.

3 (Optional) Select the *Remove Elements and Relationships Not Referenced in the Export File* check box.

This action is taken before importing the configuration file.

There could be situations in which you delete elements in one system and want to import this modified configuration to a second system, and automatically delete elements in the second system that are not in the latest imported configuration file.

Elements that are generated by adapters do not display after importing until an adapter creates them. If you have set an algorithm on an element, it should be reapplied when the adapter creates the element, because there is a reference to that element in Configuration Storage.

4 Click *OK* to import the configuration.

8.5.5 Using NOC Script to Import and Export Configuration XML Files

Use NOC Script to perform and import and export of configurations:

- "Exporting Configuration Files" on page 119
- "Importing Configuration Files" on page 119
- "Understanding XML File Structure" on page 119
- "Understanding XML Tags" on page 120
- "Element Property Tags" on page 121

Exporting Configuration Files

The following sample code exports a configuration file:

```
var server = formula.Root.findElement('formulaServer=Server/root=Administration')
var dname = element.dname;
formula.log.info( "Exporting the hierarchy starting at " + dname);
server.perform(session, "Config|ExportSilent", [], [ dname, 'd://
OperationsCenter_install_path//cstest.xml']);
```

Replace the following variables:

`element.dname`	Replace with the DName of the element to export.
`d://OperationsCenter_install_path//cstest.xml`	Replace with the path and export file name.

Importing Configuration Files

The following sample code imports a configuration file.

```
var server = formula.Root.findElement('formulaServer=Server/root=Administration')
server.perform(session, "Config|ImportSilent", [], ['d://
OperationsCenter_install_path//cstest.xml']);
```

Replace the following variables:

`d://OperationsCenter_install_path//cstest.xml`	Replace with the path and export file name.

Understanding XML File Structure

Any structure in your *Service Models* hierarchy can be translated into an XML file using the `Export` command. The file can then be edited for required changes or shared with other installations.

The well-formed XML contains basic tags and commands. At the simplest level, the XML file must contain this basic structure:

```
<views>
  <tree>
    <element>element definition tags</element>
  </tree>
</views>
```

Imported XML files need to be well-formed XML files containing necessary DTD headers, as shown in the following example XML code. This code creates a hierarchy with *Test Element* as the top element with a native child (Child Element) and a link to an existing technology branch (Tivoli TEC+):

```
<!DOCTYPE views PUBLIC "-//Managed Object Solutions, Inc.//DTD views 1.0//EN"
"http://www.ManagedObjects.com/dtds/views_1.0.dtd">

<views destroy="no">
    <tree start_at="root=Organizations">
      <element>
        <name>Test Element</name>
        <class>org</class>
        <displaySourceElements>false</displaySourceElements>
        <contact>Jim</contact>
        <company>Enigma Corp</company>
        <address>394 First Street, Fairfax, VA</address>
        <phone>703-555-1212</phone>
        <email>test@mo.com</email>
        <secure name="simple" related="yes" self="yes" children="no">
          <grant names="GroupOne,GroupTwo" permissions="view,manage,access,define"/
>
          <deny names="GroupThree" permissions="access,define"/>
        </secure>
        <element>
          <name>Child Element</name>
          <class>org</class>
          <displaySourceElements>true</displaySourceElements>
          <contact>John</contact>
          <company>Enigma Corp</company>
          <address>394 First Street, Fairfax, VA</address>
          <phone>703-555-1212</phone>
          <fax>703-555-3939</fax>
          <pager>322-325-3565</pager>
          <sref name="simple"/>
          <relate kind="ORG">script=TivoliTec+/root=Elements</relate>
        </element>
      </element>
     </tree>
</views>
```

Understanding XML Tags

Table 8-3 provides description of main XML tags used for the creation of Service Views.

Table 8-3 *Upper Level XML Tags*

Tag	Description
<views>	The initial tag provided in the XML stream that establishes the processing mode for the additional XML commands that follow. There can be only one <views> tag defined per XML file. The following attribute is specified for an <element> tag: **destroy:** Specifies if the file runs in destructive or nondestructive mode. Set it to Yes (destructive) to replace all existing elements with only those element definitions provided in the XML file. The default is No (nondestructive), which updates the properties of existing elements, retains existing elements, and updates the elements listed in the XML file.

Tag	Description
<tree>	Use within the <view> tag to specify a hierarchy of elements to build. The following attribute is specified for a <tree> tag: **start_at:** Specifies the root for the new hierarchy to enable constructing a branch at any point. If the root is not specified, the default is Organizations. For example, to modify the structure under the Northeast location, specify: <tree start_at="org=Northeast/root=Organizations">
<create>	Use within the <view> tag to add elements. If an element already exists, it is updated and a log message is generated.
<delete>	Use within the <view> tag to remove elements. The <delete> tag cannot contain nested <element> subtags.
<modify>	Use within the <view> tag to update specific elements. If the element does not exist, it is created and a log message is generated.
<update>	Use within the <view> tag to update elements while retaining an existing hierarchy. If the element does not exist, it is created and a log message is generated.
<element>	Use within the <tree>, <create>, <delete>, <modify>, or <update> tags to define elements. A complete hierarchical structure can be defined using nested <element> tags. The following attributes are specified for an <element> tag: ◆ **auto_relate_children:** Set to Yes to automatically create relationships with children elements. The default is Yes. ◆ **prevent_secure:** Set to Yes to not apply the element security to the child. The default is Yes. The <element> tag contains subtags that fully define attributes about the element including the element ID, properties, security information, and child relationships. The <name> and <class> tags are used to form the DName for the generated element. For a list of element-related tags, see Table 8-4 on page 121. Because the <update>, <create>, <delete>, and <modify> tags do not use a start_at attribute, top-level <element> tags must have a <dname> or <name>, <class>, and <parent> combination of subtags declared. Nested elements only require <name> and <class>, or a <dname> to be defined.

IMPORTANT: The <create>, <delete>, <modify>, and <update> tags do not prune the hierarchy even if view is set to destroy. They also do not change (or create) auto-related children.

Element Property Tags

The tags listed in Table 8-4 are nested within an <element> tag and specifically define the properties of the element.

Table 8-4 *Element Definition Tags*

Element Tag	Specifies…
<Lat>	The numeric value for the latitude of the location. Used specifically for elements native to the *Locations* hierarchy.

Element Tag	Specifies…
<Long>	The numeric value for the longitude of the location. Used specifically for elements native to the *Locations* hierarchy.
<address>	The mailing address for the contact person.
<algorithm>	The type of algorithm used to calculate the element's condition.
<algorithmDisseminates>	If True, child elements inherit algorithm settings. If True,StopAtNoChildren, algorithms apply to all child elements except those with no children. If True,StopAtNoDiscoveredChildren, algorithms apply to all child elements except those with no discovered children (discovery is not forced). These settings do not apply to matched children.
<algorithmParameters>	A semicolon-delimited list of algorithm parameters if required based on type specified in the <algorithm> tag.
<class>	The class of the location. If <tree start_at="root=Organizations"> is specified for the <tree> tag, the class specified for direct children (1st level) elements must be org. Otherwise, the class is not recognized by the system.
<company>	The company name for the contact person.
<contact>	The name of the contact person.
<displaySourceElements>	If True, matched elements display under the element as children. If False, matched elements do not display as children, but do contribute to the element's state.
<dname>	The name and class attributes are used to form the DName for the generated element.
<element>	Nests another element as a child under the element.
<email>	The e-mail for the contact person.
<fax>	The fax number for the contact person.
<graphic>	A graphics file to show for the element in the Layout view. Use a relative path to the image file; the path specified in the XML file is appended to the mos/html directory.
<ignoreChildAlarms>	If True, alarms from all children are not propagated to the element.
<matches>	A regular expression or class expression used to match elements.
<name>	The short name of the element.
<pager>	The pager number for the contact person.
<parent>	The DName of the parent element if the <element> tag is not nested within another <element> tag.
<phone>	The phone number for the contact person.
<propagateSecurity>	If True, security permissions from the element are inherited by matched children.

Element Tag	Specifies…
<relate>	Bind an element from any place in Operations Center by using the <relate> tag. The <relate> tag specifies a child relationship for the enclosing <element> tag. The <relate> tag accepts the following attributes: ◆ **prevent_secure:** If Yes (checked), element security is not applied to the child. ◆ **kind:** Defines the relationship type. Initially set to ORG (the default), but in the future could be set to another kind of relationship.
<script>	A script used to match elements.
<secure>	Security permissions for the element. Each <secure> tag can have the following attributes: ◆ **name:** Name of the security definition which allows you to invoke it again using the <sref> tag. ◆ **related:** If Yes (checked), applies the security permissions to all elements specified using <relate> tags, where kind=ORG. The default is No (unchecked). ◆ **self:** If Yes (checked), applies security permissions to the element itself. ◆ **children:** If Yes (checked), applies security to children of the element. ◆ **mergeRelated:** If Yes (checked), applies the grant or deny security entries to the security of the related element. If No (unchecked), the existing security is not used. The default is Yes. Within the <secure> tag, use the <grant> or <deny> tags to define actual permissions. The following attributes are specified for <grant> and <deny> tags: ◆ **names:** A comma-delimited list of users or groups who are granted or denied permission to the element. ◆ **permissions:** A comma-delimited list of permissions. The current list includes view, access, define, and manage. **TIP:** Applying security access to groups that you've defined is a more efficient method of applying security, rather than assigning security to each individual element defined within the XML file. It also enables an easier method of tracking a user's security level.
<serf>	A reference to previously defined security permissions using the <secure> tag. Specify the name of the <secure> tag to reference. Using <serf> allows security definitions to be easily reused.
<url>	A Web site address for the contact person or company.

8.5.6 Using NOC Script to Backup the Configuration to an ODB Database File

Using Operations Center scripting capability, there is a fast and easy way to export an entire configuration to an Object database (`.odb`) format. This configuration can then be saved as a backup for use if a system restore is needed, or used to copy the configuration to another Operations Center server.

The syntax for the job script is:

`formula.util.snapshotConfig( filename, directory )`

where *filename* is the name of the file to export to, and *directory* is its location.

Both parameters are optional. If *filename* is omitted, the snapshot name takes the form of `snapshot.timestamp.odb` (for example, `snapshot.20071011151144.odb`). If *directory* is omitted, the default location for files is `$Install/configstore`.

9 Administering the Operations Center Server

After Operations Center is configured, you need to administer the following servers:

- Operations Center server

  You can check the server status and system information, and run specialized scripts and jobs.
- Web server

  You can view its status information and audit options, view its condition, and start and stop it.
- Image server

  You can render dynamic and 3-D charts that include performance information using Web clients, such as the dashboard.

The following sections provide details required to administer the Operations Center server:

- Section 9.1, “Viewing Operations Center Server System Information,” on page 125
- Section 9.2, “Using Server Scripts,” on page 128
- Section 9.3, “Using Jobs,” on page 128
- Section 9.4, “Using the Web Server,” on page 137
- Section 9.5, “Using the Image Server,” on page 137

9.1 Viewing Operations Center Server System Information

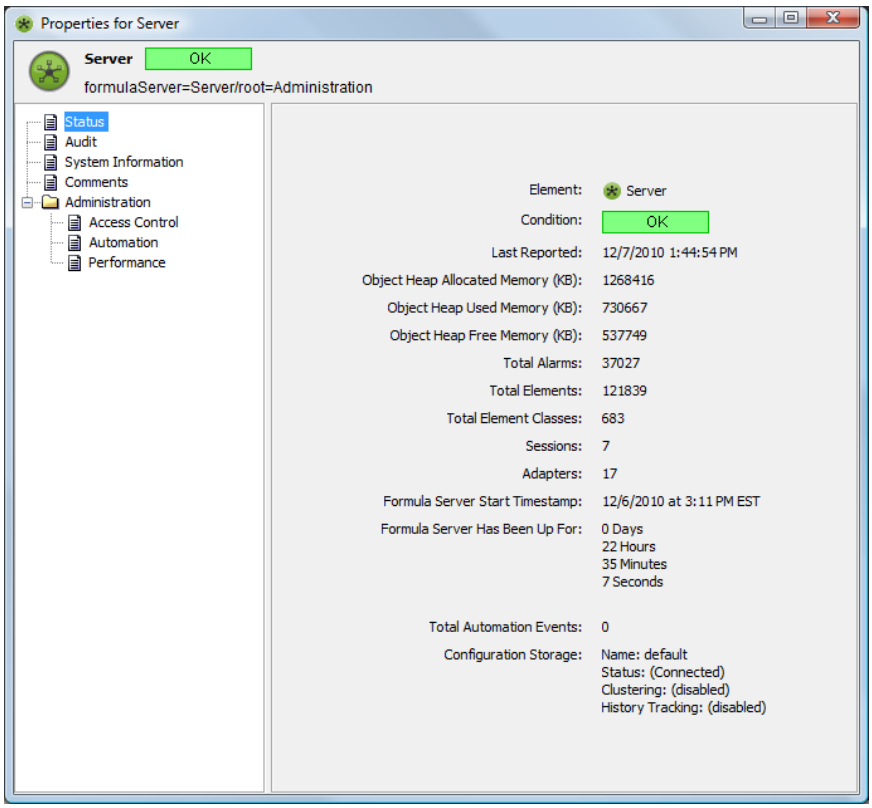
The Operations Center server displays in the Operations Center console in the hierarchy as an element:

- Section 9.1.1, “Viewing Statuses for the Operations Center Server,” on page 126
- Section 9.1.2, “Using the mosstatus Command,” on page 127

9.1.1 Viewing Statuses for the Operations Center Server

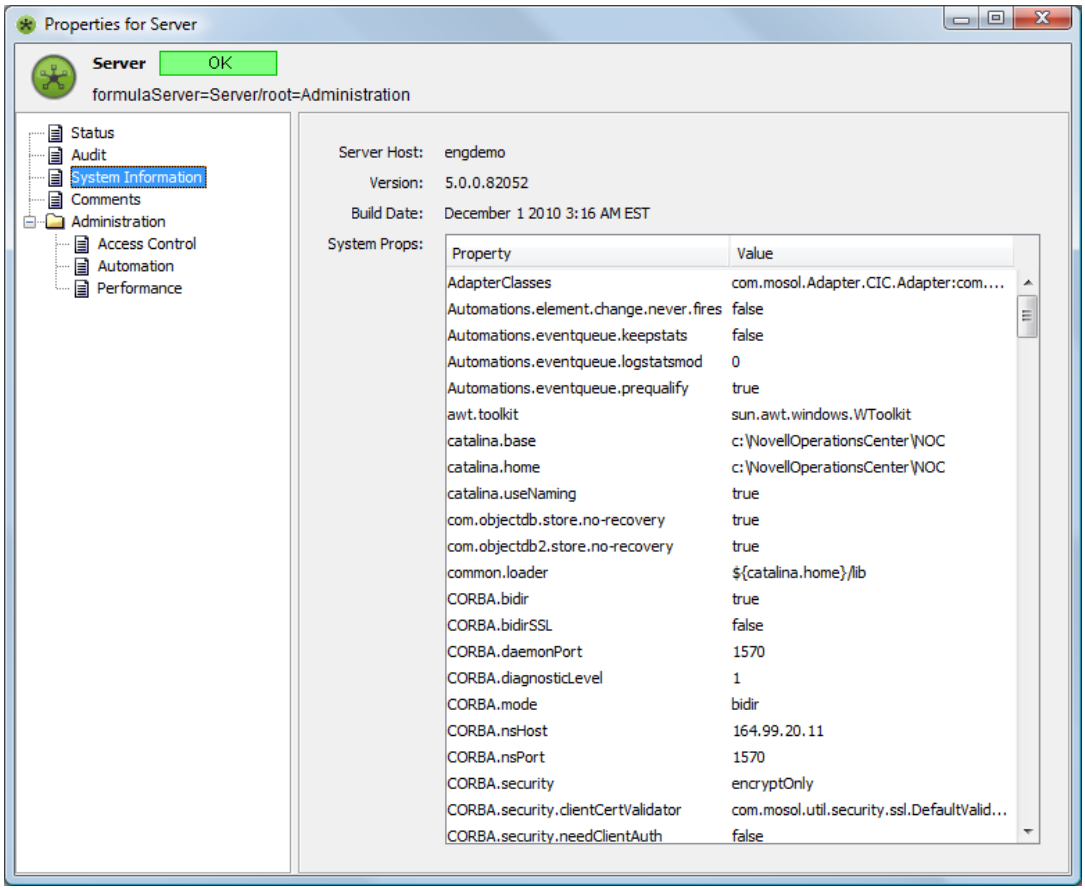
To view Operations Center server status:

1 The *Server* element is located under *Enterprise* > *Administration*. Right-click *Server*, then select *Properties* > *Status* to view the status information:



The status information shows when the server was started and the amount of time it has been running.

2 To view the system information, click *System Information* in the left pane:



The system information shows properties and their values.

3 To add and store text comments, click *Comments*.

4 To view auditing information, click *Audit*.

For more information on auditing on Operations Center server, see the *Operations Center 5.5 Security Management Guide*.

9.1.2 Using the mosstatus Command

The `mosstatus -all` command provides the following information:

- Verify that the Operations Center software was successfully installed and is running
- Names and status of all adapters that are currently running
- Number and type of active sessions
- License expiry date
- Uptime for the Formula server software

- Version of the Formula server software
- Server Patches installed

To use the `mosstatus` command:

From the *`drive`*`:/OperationsCenter_install_path/bin` directory, enter `mosstatus -all` at the command prompt, where *drive* is the installation drive.

9.2 Using Server Scripts

On the Operations Center server, you can run scripts to perform specific tasks based on actions that you want to take for specific server states.

Scripts are created using NOC Script, a scripting language that allows customizing the capabilities of the Operations Center server for more effective management. For more information about NOC Script, see the *Operations Center 5.5 Scripting Guide*.

To run scripts on the Operations Center server when it starts, initializes, stops, or fails:

1. Open the */OperationsCenter_install_path*`/database/shadowed/Adapters.ini` file.
2. Add the following commands into the `[Formula]` section:

Command	Function
`Script.onStarted=@("`*`script_name`*`.fs)";`	Runs the script's code when the server starts.
`Script.onStopped=@("`*`script_name`*`.fs)"`	Runs the script's code when the server stops.
`Script.onInitialized=@("`*`script_name`*`.fs)";`	Runs the script's code when the server initializes.
`Script.onError=@("`*`script_name`*`.fs)";`	Runs the script's code when the server fails.

9.3 Using Jobs

You can use a job to run scripts. Jobs, which are configured and managed in the Operations Center console, can be run, scheduled, enabled and disabled, and stopped. You can also view the status of jobs and change the definition of jobs.

Review the following sections for how to use jobs in Operations Center:

- Section 9.3.1, "Accessing Jobs," on page 129
- Section 9.3.2, "Scheduling Jobs," on page 129
- Section 9.3.3, "Running Jobs," on page 134
- Section 9.3.4, "Viewing the Status of Jobs," on page 134
- Section 9.3.5, "Enabling and Disabling Jobs," on page 135
- Section 9.3.6, "Stopping Jobs," on page 136
- Section 9.3.7, "Changing Job Definitions," on page 137

9.3.1 Accessing Jobs

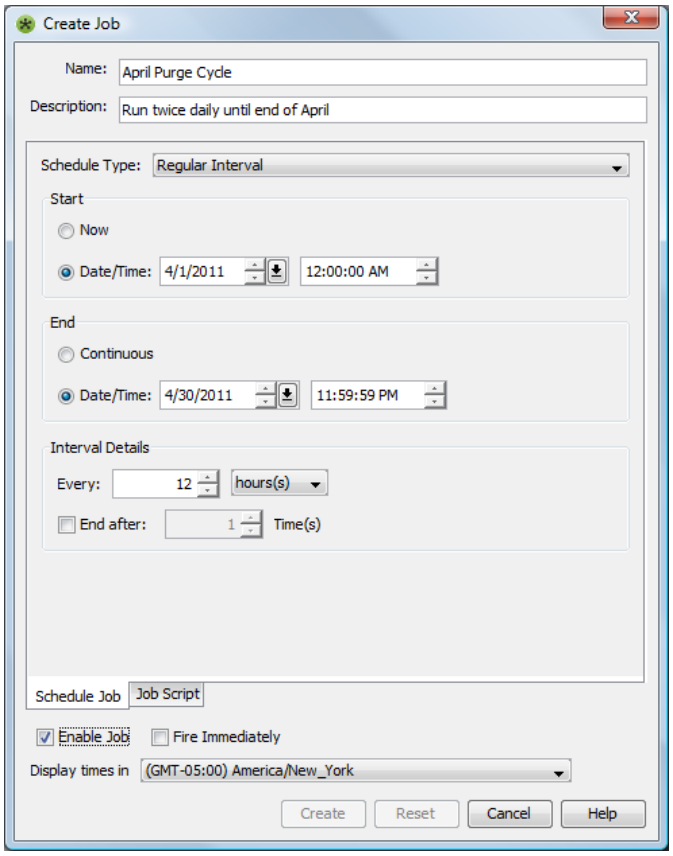
To view job definitions, in the Operations Center console under *Enterprise*, click *Administration* > *Time Management* > *Jobs*. An element for each job definition displays.

9.3.2 Scheduling Jobs

Scheduling a job requires defining when it will run, associating a script to run, and enabling the job.

To schedule a job:

1. In the *Explorer* pane, expand *Administration* > *Time Management*.
2. Right-click *Jobs*, then select *Create Job*.



3. Click the *Schedule Type* drop-down list, then select the job's schedule type:

   Cron String: Runs the job based on the time interval specified by a raw cron string.

   For more information on cron strings, see "Cron String" on page 132.

   Regular Interval: Runs the job at a particular time interval for an optional number of times.

   Specify one of the following as a start time:

   - **Now:** To start the job immediately.
   - **Date/Time:** A specific start date and time.

   Specify one of the following for an end date/time:

   - **Continuous:** No end date/time.

- **Date/Time:** A specific end date and time.

Select an interval (minutes, hours, months, and so on).

End the job after a certain number of runs by specifying the number of times to run the job.

Daily Interval: Runs the job every day or every x number of days, at the specified time interval.

Specify one of the following as a start time:

- **Now:** To start the job immediately.
- **Start Date:** A specific start date and time.

Specify one of the following as an end date:

- **None:** No end date.
- **End Date:** A specific end date.

Specify how often to run the job during a day:

- **Once:** Runs the job once as the specified start time.
- **Every:** Runs the job at the specified time interval (seconds, minutes, hours).

Specify the daily interval to run the job. For example, enter 5 to run the job every five days at the specified frequency.

Weekly Interval: Runs the job on a weekly basis, on particular days of the week.

Specify a start and end date.

Specify a daily frequency as described above for the Daily Interval schedule.

Select the weekly interval by entering a number. For example, enter 2 to run the job every two weeks.

Select the days of the week when the job runs. Use *Work Week* to select Monday through Friday; use *Weekend* to select Saturday and Sunday; or, select *All*.

Monthly Interval: Runs the job on a monthly basis.

Specify a start and end date.

Specify a daily frequency as described above for the Daily Interval schedule.

Select one of the following for the monthly interval:

- **Day x of every y month:** Runs the job on a specific day of the month (enter 1 for the first day) of every x month. For example, enter 3 to run every three months.
- **The x y of every z month:** Runs the job on the first, second, third, fourth or last day (or specific day) of every z months. For example, specify the last Friday of every fourth month.

4 Complete the options appropriate for the schedule type selected.

If the *Regular Interval* schedule type is selected with a start date/time, consider selecting the *Fire Immediately* check box to run the job as soon as it is created, rather than waiting for the next date/time interval before starting.

If you do not fire the script immediately, the script starts at the next specified interval after the Operations Center server starts.

5 Select a time zone from the *Display Times In* drop-down list.

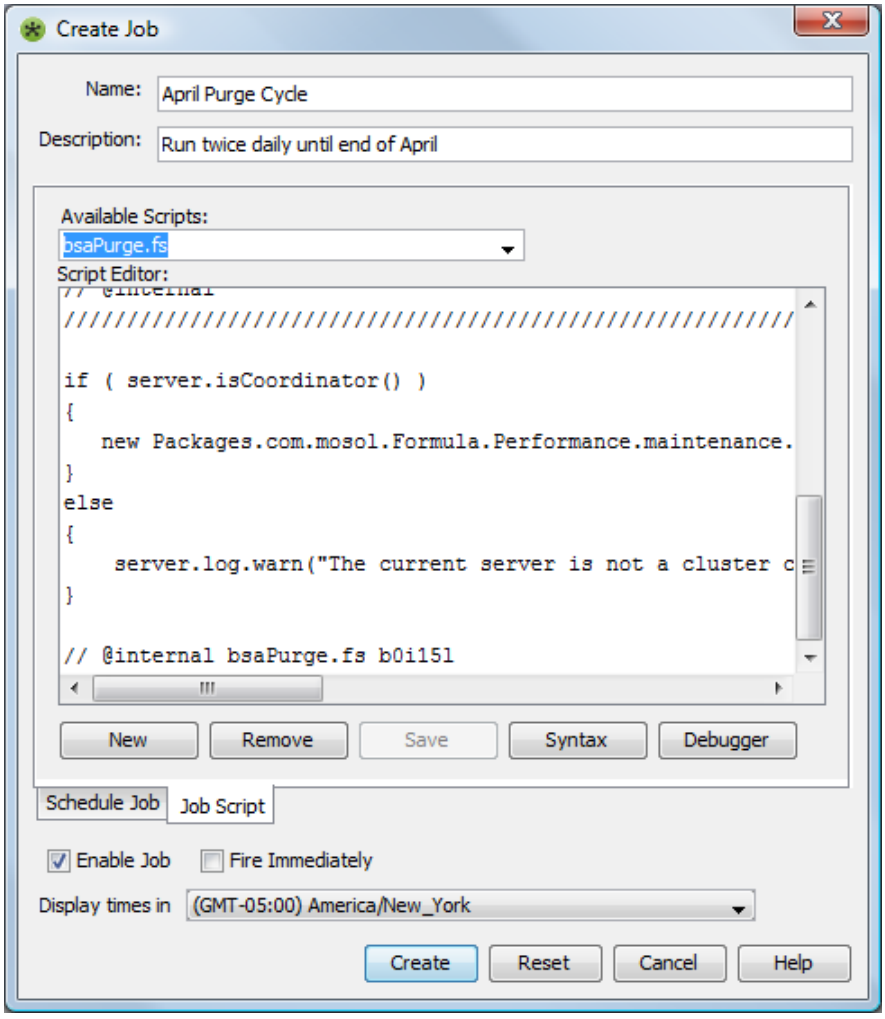
Either select the time zone for starting and ending the job, or use the local host time for either the Operations Center server or the client.

If the job can be executed from servers in different time zones, consider selecting *Server Default Time Zone*, which uses the local server's time zone (as defined by its operating system).

Select the *Do Not Adjust Execution Time for Time Zone* check box to use the job's specified start and end times based on the local server's time zone, and not add or subtract hours when the job restarts on a server with a different time zone. Leave this check box unmarked to adjust the start and end times when the server time zone changes.

6 Click the *Job Script* tab to specify the script to run.

When defining the script that will run for the job, either select an existing script or enter a new one. The dialog box in which you enter a script has options for checking the syntax of the script and debugging the script.



7 Perform one of the following steps:

- Click the *Available Scripts* drop-down list, then select the script file to use.

  The script code displays in the Script Editor section. Make any changes in the script code in the Script Editor section.

- Click the *New* button to enter a new script, then:

  1. Enter the name of the new script file in the *Available Scripts* drop-down list.
  2. Click the *Syntax* button to have Operations Center software check the syntax of the script.
  3. Click the *Debugger* button to debug the script.
  4. When complete, click the *Save* button to add the new script.

8 Select the *Enable Job* check box to activate the script when the server starts.

9 Click the *Create* button to save the job.

Cron String

You can specify the interval that a job will run as a time interval specified by using a raw cron string. A cron string is an expression-type command that is comprised of six or seven field values separated by white spaces. These fields are described in Table 9-1.

Table 9-1 *Cron String Fields*

Field Name	Values	Special Characters
Seconds	0–59	, - * /
Minutes	0–59	, - * /
Hours	0–23	, - * /
Day of Month	1–31	, - * ? / L C
Month	1–12 or JAN–DEC	, - * /
Day of Week	1–7 or SUN–SAT	, - * ? / L C #
Year (optional)	empty, 1970–2099	, - * /

Table 9-2 describes the characters that can be used in the cron string. The legal characters and the names of months and days of the week are not case sensitive.

Table 9-2 *Cron String Special Characters*

Name	Character	Description
Asterisk	*	Indicates all possible values. For example, as a value in the *Minutes* field runs every minute.
Question Mark	?	Indicates no specific value. Available for *Day of Month* and *Day of Week*. This is useful to specify something in one of the two fields, but not the other. See the examples in Table 9-3 on page 133 for clarification.
Hyphen	-	Specifies a range. For example, 10–12 in the *Hours* field runs for hours 10, 11, and 12.
Comma	,	Declares additional values. For example, MON,WED,FRI in the *Day of Week* field runs the job on Monday, Wednesday, and Friday.
Forward Slash	/	Defines increments. For example, 0/15 in the *Seconds* field runs the job on seconds 0, 15, 30, and 45. Whereas, 5/15 in the *Seconds* field runs the job on seconds 5, 20, 35, and 50. Can be used after the * (asterisk) character where the * is interpreted as 0.

Name	Character	Description
Capital L	L	Specifies last, but has different results depending on the field. Available for *Day of Month* and *Day of Week*. For example, in the *Day of Month* field runs the last day of the month (day 31 for January, day 28 for February on nonleap years). If in the *Day of Week* field by itself, it runs on 7 or SAT. Use in the *Day of Week* field after another value to run the last X day of the month (6L is the last Friday of the month). Do not specify lists, or ranges of values, as there could be confusing results.
Pound Symbol	#	Specifies the nth day of the month. Available for *Day of Week* only. For example, 6#3 in the *Day of Week* field runs the third Friday of the month (day 6 = Friday and "#3" = the 3rd one in the month). 2#1 runs the first Monday of the month and 4#5 the fifth Wednesday of the month. Note that if #5 is specified and there is no 5th day of the given day-of-week in the month, then no firing occurs that month.
Capital C	C	Indicates the value is evaluated against an assigned calendar, if any. Available for *Day of Month* and *Day of Week* fields. If no calendar is associated, then it is equivalent to having an all-inclusive calendar. For example, 5C in the *Day of Month* field runs the first day included by the calendar on or after the 5th. Whereas, 1C in the *Day of Week* field runs the first day included by the calendar on or after Sunday.

Table 9-3 shows a few examples of full cron string expressions.

Table 9-3 *Cron String Examples*

String	Job Fires...
0 0 12 * * ?	At 12pm (noon) every day
0 15 10 ? * *	At 10:15am every day
0 15 10 ? * *	At 10:15am every day
0 15 10 * * ?	At 10:15am every day
0 15 10 * * ? 2005	At 10:15am every day during the year 2005
0 * 14 * * ?	Every minute starting at 2pm and ending at 2:59pm, every day
0 0/5 14 * * ?	Every 5 minutes starting at 2pm and ending at 2:55pm, every day
0 0/5 14,18 * * ?	Every 5 minutes starting at 2pm and ending at 2:55pm, and fire every 5 minutes starting at 6pm and ending at 6:55pm, every day
0 0-5 14 * * ?	Every minute starting at 2pm and ending at 2:05pm, every day
0 10,44 14 ? 3 WED	At 2:10pm and at 2:44pm every Wednesday in the month of March
0 15 10 ? * MON–FRI	At 10:15am every Monday, Tuesday, Wednesday, Thursday, and Friday
0 15 10 15 * ?	At 10:15am on the 15th day of every month

String	Job Fires...
0 15 10 L * ?	At10:15am on the last day of every month
0 15 10 ? * 6L	At 10:15am on the last Friday of every month
0 15 10 ? * 6L	At10:15am on the last Friday of every month
0 15 10 ? * 6L 2002–2005	At 10:15am on every last Friday of every month during the years 2002, 2003, 2004, and 2005

9.3.3 Running Jobs

After enabled, jobs run according to the schedule defined for each job.

A job can also be manually run outside a defined schedule without interrupting its scheduled runs.

To manually run a job outside its schedule:

1. In the *Explorer* pane, expand the root *Administration* element > *Time Management* > *Jobs*.
2. Right-click the job definition, then select *Run Job*.

9.3.4 Viewing the Status of Jobs

Table 9-4 lists the various conditions that a job element can have.

Table 9-4 *Job Conditions*

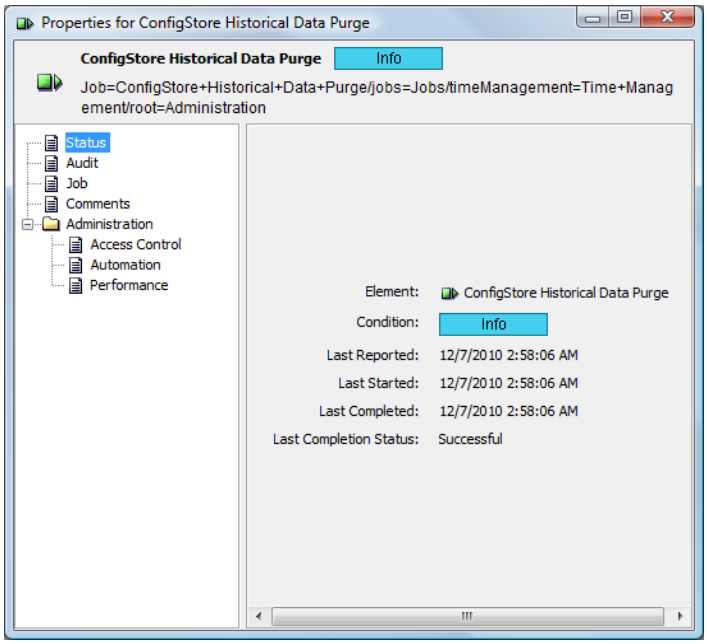
Default	Condition	Description
	INFORMATIONAL	Job is scheduled to run, but is currently dormant
	OK	Job is in the process of being run
	MINOR	Job is enabled, but completed
	UKNOWN	Job is disabled
	CRITICAL	Job failed

To view a job's status:

1. In the *Operations Center* hierarchy under *Enterprise*, click *Administration* > *Time Management* > *Jobs*.
2. Do one of the following:
   - Mouse over the *Jobs* element to view a summary of the state of all jobs.
   - Mouse over a specific job to learn about its current state.

3 To learn more about a particular job, right-click the job, select *Properties*, then in the Properties dialog box, select *Status*.

The dialog box displays the condition, as well as the last reported, started, and completed times, and the last completion status:



9.3.5 Enabling and Disabling Jobs

You can enable or disable a specific job, or all jobs:

- "Enabling or Disabling a Specific Job" on page 135
- "Enabling or Disabling All Jobs" on page 135

Enabling or Disabling a Specific Job

To enable or disable a job:

1 In the *Explorer* pane, expand the root *Administration* element > *Time Management* > *Jobs*.

2 Right-click the job definition, then select *Enable Job* or *Disable Job*.

Enabling or Disabling All Jobs

To enable or disable all jobs:

1 In the *Explorer* pane, expand the root *Administration* element > *Time Management*.

2 Right-click the *Jobs* element, then select *Enable All Jobs* or *Disable All Jobs*.

9.3.6 Stopping Jobs

You can stop jobs manually or from within a script:

- "Manually Stopping a Job" on page 136
- "Using a Script to Stop a Job" on page 136

Manually Stopping a Job

To stop a job in the Operations Center console:

1. In the *Explorer* pane, expand the root *Administration* element > *Time Management* > *Jobs*.
2. Right-click a job, then select *Stop Job*.

   A message displays indicating the job is stopped.

   The *Stop Job* option attempts to interrupt the selected job while it is running. In some cases, the Operations Center server might not be able to halt the job. If the job cannot be stopped, the following message displays:

   ```
   Attempted to interrupt/stop job, but the job did not respond.
   ```

Using a Script to Stop a Job

When creating a script, include an operation that periodically detects the interrupt flag set by selecting *Stop Job* option on the job element.

If the script detects this interrupt flag, it should exit the job. Any script or class can check the interrupt flag and exit if it has been interrupted.

For information on creating scripts, see Section 9.2, "Using Server Scripts," on page 128.

The following sample script shows how to periodically check for an interrupt by the `Stop Job` command:

```
var i = 0;
while(true)
{
   if ( i == 20000000 )
   {
      i = 0
      formula.log.error('checking for interruption!')
      if ( java.lang.Thread.interrupted() == true )
      {
        formula.log.error('Detected interrup…stopping!')
        break;
      }
   }
   i++
}
```

9.3.7 Changing Job Definitions

To modify or delete a job definition:

1. To edit the job, do the following:
   - **1a** Right-click the job, then select *Properties*
   - **1b** In the Properties dialog box, select *Job*. The Job definition properties display.
2. To delete a job, right-click it, then select *Delete Job*.

9.4 Using the Web Server

The Web server appears in the Operations Center console as an element.

To access the Web server:

1. In the hierarchy under *Enterprise*, expand *Administration* > *Server* > *Web Server*.
2. Do any of the following:
   - For options to start and stop a Web server, right-click it.
   - To view status information and audit options, right-click *Web Server* to access the Properties dialog box.
   - To view the condition of a Web server, right-click *Web Server* to access the Properties dialog box. Status information displays.
3. If security concerns make it necessary to disable HTTP TRACE requests to the Web server, for further instructions see "Configuring Communications Security " in the *Operations Center 5.5 Security Management Guide*.

9.5 Using the Image Server

The Image server allows Operations Center Web clients (including the dashboard) to render dynamic and 3-D charts that include performance information.

Use the Image server console to do the following:

- Stop or start the Image server
- Clear the Image server cache
- View performance statistics
- Turn on debug or logging
- Set security permissions (important in a multi-homed server environment)
- Change the admin password for the Image server console
- Set memory settings
- Set caching and communication settings

To use the Image server console:

1. To access the Image server console, from a Web browser, enter the following URL in the *Address/Location* field:

   http://*hostname*:3004/casapp/administrator

   where *hostname* is the hostname of the Image server.

The paths for the Image server are not configurable using Configuration Manager after installation or upgrade of the Operations Center software. The Image server paths are created during installation and can only be modified using the Corda administrative tools.

Corda user documentation is not distributed with Operations Center, so the Help links from the Image Server Administration console are not valid. To access Corda documentation online, go to http://www.corda.com/devzone/docs (http://www.corda.com/docsource/doc6/server/docs/pdfhelp.html).

IMPORTANT: If you change the hostname or installation location after installation, the Image server paths must be updated using the Corda administrative tools.

You must create a new entry for any valid IP address and/or hostname for every URL to access the Operations Center server using the *Path/URL Permissions* option.

2 Enter the appropriate password in the *Password* field.

The default password is `formula`. If this has been changed, enter the new password.

3 Click *Submit*.

The Server Administrator page opens and displays a list of current settings.

4 (Conditional) To create a new entry for any valid IP address or hostname for every Image server URL to be able to access the Operations Center server, click *Path/URL Permissions* under SECURITY.

5 Update the Image server options as required. Table 9-5 lists the options available in the Image server console that relate to managing the Image server.

Table 9-5 *Image Server Options*

Section	Option	Description
SERVER SETTINGS	*Address / Port*	This option is NOT used to set Image server ports. Use Configuration Manager to set these.
	Cache / Connections	Set cache and connections settings as necessary for performance requirements.
		Cache Size specifies the size (in number of images) of the Server cache. It is recommended that this number be between 100 and 500 images.
		Maximum Connections specifies the maximum number of requests that can connect to server simultaneously. Other requests are queued and served in the order they arrive.
	Memory Settings	Adjust the amount of memory allocated to the server. If routinely charting large data sets or receiving errors from Dashboard charts, the recommended memory setting is 128KB or greater.
	Advanced Settings	Set levels of image quality for rendering charts. If using JPEG output, consider increasing the quality setting.
PERFORMANCE	*Statistics*	View statistics including total hits, memory in use, max memory available, and maximum threads open.
	Charts Served	View server up time, number of images served and server status. View graphical charts showing server hits by hour, by day, and by month.

Section	Option	Description
DEBUGGING & LOGGING	*Debug Settings*	Enable debugging with normal or verbose settings.
	Logging Settings	Enable logging and set number of transactions to log.
SECURITY	*Path/URL Permissions*	Set allowable URLs for Image server, including any valid IP addresses or hostnames. If operating in a multi-homed server environment, add all additional servers to the PathMaps code.
	Change Password	Change the password for the Image Server Administration console only.

6 To manage the Image server, click *Home* and do any of the following:

- To stop the Image server, click the *Stop* button.
- To restart the Image server, click the *Restart* button.
- To clear the Image server cache, click the *Clear Cache* button.

Customizing the Implementation

Administrators and users alike will use the Operations Center console to monitor various aspects of your business and services, which are represented as elements and alarms. State and performance information is then analyzed to provide information as to the current health of these monitored elements.

As an Operations Center administrator, you can customize how and what information is available and exposed to your users. Depending on the permissions granted to users, they can perform configurations and other tasks.

All users have the ability to customize and change view options in the Operations Center console. The *Operations Center 5.5 User Guide* covers basic functionality available in the Operation Center console and those features normally available to all users.

This sections covers the following topics that will allow you to customize your implementation:

- Chapter 10, “Enabling and Disabling Console Functionality,” on page 143
- Chapter 11, “Customizing Monitored Elements and Alarms,” on page 147
- Chapter 12, “Capturing Alarm and Performance History,” on page 177
- Chapter 13, “Time Categories, Calendars, and Schedules,” on page 197
- Chapter 14, “Defining and Managing Automation Events,” on page 213
- Chapter 15, “Using Algorithms to Calculate Element State,” on page 237

10 Enabling and Disabling Console Functionality

- Section 10.1, “Customizing the Console to Restore Windows,” on page 143
- Section 10.2, “Overriding the Maximum Elements Limit for the Network View,” on page 143
- Section 10.3, “Disabling the Network and Layout Views,” on page 144
- Section 10.4, “Configuring Element Find Features,” on page 144
- Section 10.5, “Enabling Chat,” on page 146

10.1 Customizing the Console to Restore Windows

One of the disadvantages of using the standard *Tile* and *Cascade* options to toggle between multiple windows is that the sizing and position of windows that you have configured are not maintained.

The Operations Center administrator can use the `browserframe.allowrestore` setting so that configurations are restored when window minimize and maximize buttons are used to maximize one window, then return to the original multiple window configuration by clicking that window's minimize button.

To retain the configuration of windows after one window is maximized:

1. Edit the */OperationsCenter_install_path*`/html/client/template/launch.jnlp` file, then add the following property assignment in the `resources` section:

   ```
   <property name="browserframe.allowrestore" value="true"/>
   ```

2. Run the Operations Center Configuration Manager and click *Apply*.

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

10.2 Overriding the Maximum Elements Limit for the Network View

By default, the Network view enforces a maximum of 1,000 elements. Override this limit by editing the `launch.jnlp` file.

To change the element limit for the Network view:

1. Add the following property to the `template/launch.jnlp` file, and set the value to a number greater than zero:

   ```
   <property name="jnlp.BSV.Layout.ElementLimit" value="max_number_of_elements" /
   >
   ```

2 Run the Operations Center Configuration Manager and click *Apply* to register the new setting. Then relaunch the Operations Center console.

   For instructions, see Section 2.1, "Accessing and Using Configuration Manager," on page 17.

10.3 Disabling the Network and Layout Views

Performance issues in some client environments might require disabling the *Network* and *Layout* views in the Operations Center console for nonadmin users.

These views are disabled by changing the current Operations Center console configuration directly through the `/OperationsCenter_install_path/html/applet_params.xml` file.

You can create a custom applet parameters file that is not overwritten during installation. Saving custom parameters in a separate file called `applet_params.custom.xml` guarantees that you need not reapply them after installation.

For more information on creating a custom `applet_params.xml` file, see the *Operations Center 5.5 Server Installation Guide*.

To disable the *Network* and *Layout* views:

1 To update the `applet_params.xml` (or `applet_params.custom.xml`) for the *Network* and *Layout* views, change the XML code to the following:

```
<param name="SVG.ShowLayoutSVG" value="false" />
<param name="BrowserFrame.ShowNetworkPanel" value="false" />
<param name="BrowserFrame.ShowAllPanelsToAdmins" value="true" />
```

   When `SVG.ShowLayoutSVG` and `BrowserFrame.ShowNetworkPanel` are set to False, users who belong to the administrators group can see the *Network* and *Layout* view tabs in the Operations Center console. Other users do not see these tabs and cannot access these views.

2 To allow users in the administrators group to access the *Layout* and *Network* views through the console, set `BrowserFrame.ShowAllPanelsToAdmins` to True.

   For more information on the administrators group and permissions, see the *Operations Center 5.5 Security Management Guide*.

10.4 Configuring Element Find Features

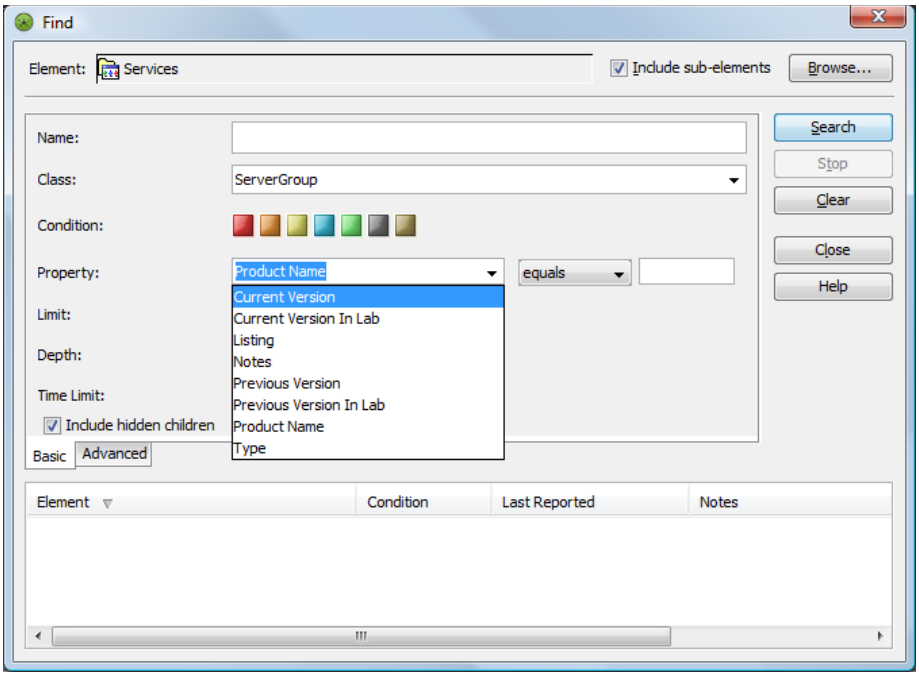
The following sections cover features that can be enabled in Find, as well as instructions for disabling Find for users:

- Section 10.4.1, "Enabling Metamodel Classes or Properties as Search Criteria," on page 145
- Section 10.4.2, "Disabling the Find Feature," on page 145

10.4.1 Enabling Metamodel Classes or Properties as Search Criteria

If the Administrator has enabled enhanced features to find elements based on specific metamodel classes or custom properties, an enhanced *Basic* tab displays, as shown in Figure 10-1:

Figure 10-1 *Enhanced Basic Tab for Find Feature*



You can specify search criteria for classes and properties, or click the arrow button, then select predefined custom class or property.

When typing search criteria in the *Class* or *Property* fields, standard * wildcard can be applied. For example, `*Class*` matches `MyClassName` and `YourclassName`.

In addition, you can filter elements based on multiple condition levels. By default, all conditions are included. Click a condition color to exclude elements that have that condition from the "found" list.

Instructions for enabling these enhanced Find features are in "Using Classes and Properties to Enhance Finding Elements" in *Operations Center 5.5 Service Modeling Guide*.

10.4.2 Disabling the Find Feature

Disabling the Find feature disables it for all users and groups.

To disable find:

1. Add the following line to the `/OperationsCenter_install_path/html/client/template/launch.jnlp` file:

   ```
   <property name="MainFrame.findEnabled" value="false" />
   ```

2. Run the Operations Center Configuration Manager and click *Apply*.

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

10.5 Enabling Chat

The Chat feature allows users to communicate with each other through the Operations Center console. In order to enable chat, the user or group must be given View, Manage, and Access permissions to the Sessions object.

NOTE: The Chat feature available via the Operations Center console allows communication between console users only. Chat is not available to dashboard users via console chat.

To enable chat:

1. From the *Explorer* pane, expand *Administration* > *Security* > *Access Control*.
2. Under Access Control, expand *Administration* > *Server* > *Sessions*.
3. Right-click *Sessions*, then select *Properties* to open the *Status* property page.
4. In the left pane, click *Administration* > *Access Control* to open its property page.
5. Click the *Entry for User/Group* drop-down list, then select the desired user or group.
6. Select the *View*, *Manage*, and *Access* check boxes.
7. Click *Set*.

   The permissions are added to the *Access Control Entries* list.
8. Click *Apply* to save the changes.

11 Customizing Monitored Elements and Alarms

There are different ways to customize the elements and alarms displayed in the Operations Center console. Elements originate from different sources, but it is possible to change their appearance in the Console to facilitate analysis and reporting.

A simple customization is to change the icons used to represent elements in the Console. Another useful feature is to create custom element and alarm menu options that are attached to specific objects or alarms based on criteria matching and can perform scriptable actions in the Operations Center Console or on the server.

To customize alarm management, administrators can specify which alarm columns display in the Console and the order in which they display. Configuring elements and alarms for suppression and acknowledgement helps administrators to troubleshoot issues related to alarms, or to identify how a new event affects the state of an element.

Other areas of controlling alarm display involve setting the maximum number of alarms displayed in the Console and limiting the roll-up of alarms to root elements, for performance reasons.

The following sections explain how to customize monitored elements and alarms:

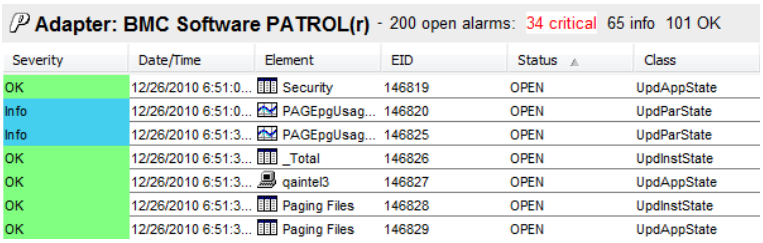
- Section 11.1, “Customizing Icon Display,” on page 148
- Section 11.2, “Determining an Element’s Class Name,” on page 150
- Section 11.3, “Displaying Element Availability Time,” on page 150
- Section 11.4, “Modifying Element and Alarm Menus,” on page 151
- Section 11.5, “Filtering Alarm Columns,” on page 156
- Section 11.6, “Configuring Suppression and Acknowledgement,” on page 157
- Section 11.7, “Controlling Display of Alarms,” on page 163
- Section 11.8, “Filtering Service Model Alarms,” on page 166
- Section 11.9, “Managing Administration Elements on Remote Servers,” on page 174

11.1 Customizing Icon Display

Elements originating from OpenView or NetView networks display with the standard icons used by the management system. Other elements display with icons from the Operations Center Icon Library based on the element's class.

Customized icons can be assigned for any non-system object class to replace the default icon. Custom icons display anywhere the element is represented in the Operations Center console. This includes the *Element* column in the *Summary* and *Alarms* views.

Figure 11-1 *Element Icons Displayed in Alarms View*



To assign a new icon to a class of elements:

1. In the *Explorer* pane, expand *Administration* > *Metamodel* > *Classes* > *Custom*.

   Open the section that contains the class for which you want to assign a new icon to update the *View* pane.

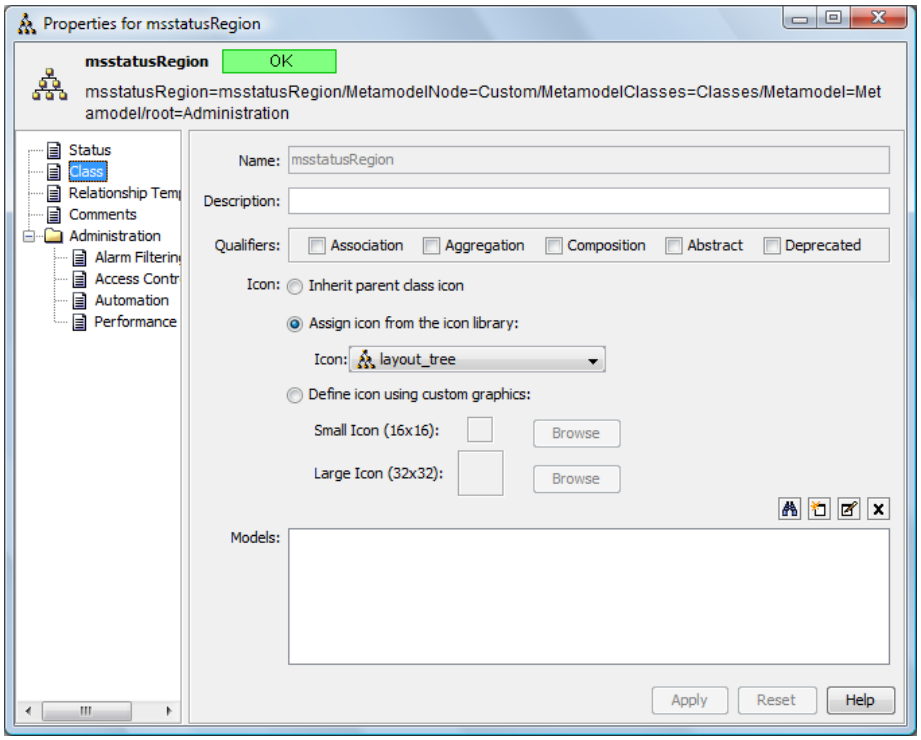
2. Do one of the following:

   - Right-click the class and select *Properties*. When the Properties dialog opens, select the *Class* tab.
   - Select the class, then open the *Portal* view. Click the *Down Arrow* for the *Class* box.

3 To assign a new icon, do one of the following:

- Select the *Assign Icon from the Icon Library* radio button, click the *Icon* drop-down list, then select the existing icon to display for the class.
- Select the *Define Icon Using Custom Graphics* radio button (Small Icon or Large Icon), click *Browse*, navigate to the icon, then click *Open*.

  The selected icon displays on the Create Class dialog box.



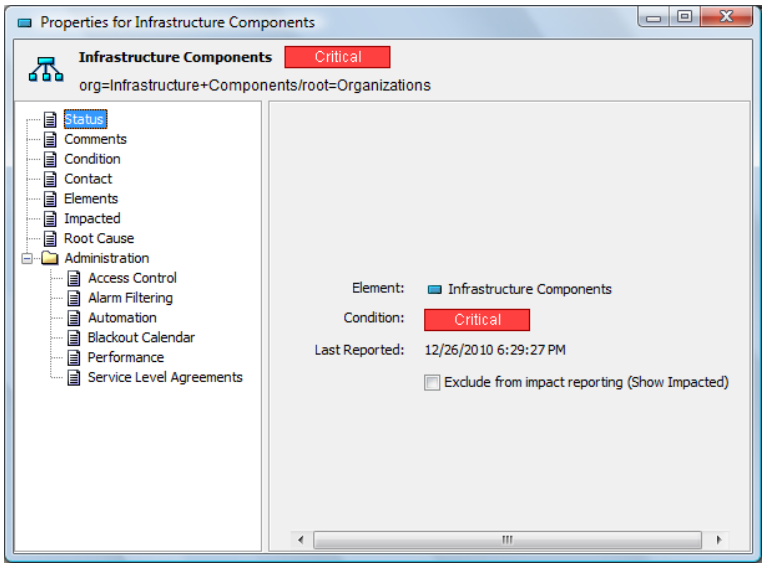
4 Click *Apply* to display the new icon for the class.

11.2 Determining an Element's Class Name

The class name usually represents a type of resource. To change the icons that display for elements, or a layout object, it is necessary to identify the class name.

To determine an element's class name:

1. In the *Explorer* pane, expand *Elements*, then select an element.
2. Right-click the element, then select *Properties* to open the *Status* property page:



The description under the element name (at the top of the property page) specifies the class as the first parameter of the Dname. In the illustration above, the element is of class *org*.

11.3 Displaying Element Availability Time

The *Simple Uptime* option can be added to any element to display how long an element has been available or unavailable, based on its current state and its immediately prior states. Simple Uptime includes all states other than UNKNOWN and CRITICAL. Simple Downtime includes UNKNOWN and CRITICAL states only.

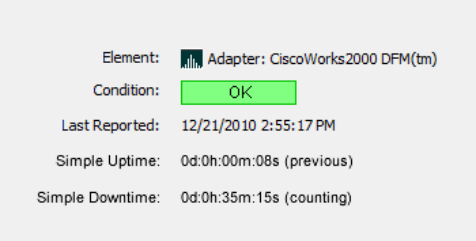
To enable the *Simple Uptime* option:

1. Enter the following code into the `Formula.custom.properties` file:

```
#
# Simple Uptime
#
# This adds information to the primary property sheet that
# shows how long the object has been available or how long
# it has been unavailable.
#
SimpleUptime=true
```

When enabled, two time stamp values are displayed for each element:

- **Simple Uptime:** Elapsed time the element has been in a nonCRITICAL/UNKNOWN state, or the elapsed time it was in a nonCRITICAL/UNKNOWN state before becoming CRITICAL/UNKNOWN.
- **Simple Downtime:** Elapsed time the object has been in a CRITICAL/UNKNOWN state, or the elapsed time it was in a CRITICAL/UNKNOWN state before becoming nonCRITICAL/UNKNOWN.



An indicator displays after each field to indicate if the state reported is the current state (counting), or the states prior to the current element state (previous).

11.4 Modifying Element and Alarm Menus

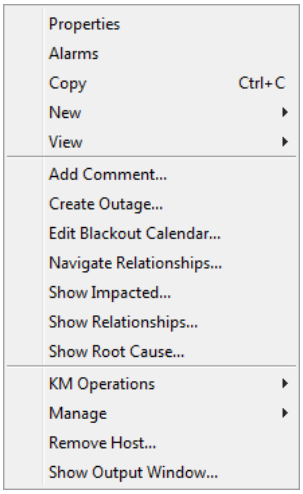
Custom options can be created for element or alarm menus. These menu options can be attached to specific objects or alarms based on criteria matching and can perform scriptable actions in the Operations Center console, or on the server. Possible options include:

- creating trouble tickets by selecting one or more alarms in an Alarms dialog box
- Running a program on the server that copies important alarms to a SQL database
- Displaying online help for a particular piece of equipment or rule name within an alarm
- Launching an element-specific management system on the user's workstation for a particular class of managed element

By default, options display in groups in the menu as follows:

- **Navigation:** Contains options open the Properties dialog box and to jump to different views.
- **System:** Contains system operations.
- **User Defined:** Contains user-defined operations. Add new menu options the Create Operation dialog box in the Operations Center console.

Figure 11-2 *Right-Click Menu Example. Navigation menus show in the upper section. System menus show in the middle section, and user defined menus show in the bottom section.*



The following sections explain how to modify alarm and element menus:

- Section 11.4.1, “Creating New Operations Through the Operations Center Console,” on page 152
- Section 11.4.2, “Adding New Menu Items in the Operations.ini File,” on page 154
- Section 11.4.3, “Adding Colored Menu Operations,” on page 155
- Section 11.4.4, “Optimizing Alarm Selection,” on page 155

11.4.1 Creating New Operations Through the Operations Center Console

To create a custom operation:

1. In the *Explorer* pane, expand *Administration* > *Server*.
2. Right-click *Operation Definitions*, then select *Create Operation* to open its dialog box.
3. Define the operation using the parameters defined in the following fields:

   Name: Name of the menu definition. This does not display in the Operations Center console.

   Menu Text: The option name to display on the menu.

   Context: Select *Element* or *Alarm* to identify the object type associated with the new menu option. Select *Both* to associate the menu option with matching alarms and elements.

Match By: Attaches the new menu option to an element or alarm menu if the following conditions are met:

- **Distinguished Name (DName):** Element DName matched by the end of the string. To match an element exactly, specify the element's full DName. As DNames are segmented from specific to general, you can match groups of elements by specifying part of the distinguished name.
- **Distinguished name expression:** A combination of DName and regular expression that can be used to further define the name of the element or alarm. A regular expression is a pattern that matches a set of strings. Regular expressions can be constructed by using various operators to combine smaller expressions.
- **Name:** The label given to the element or alarm.
- **Name expression:** A combination of a given name and regular expression to be used to further define a matching name of an element or alarm.
- **Class name:** Element class name that attaches the new menu option to an entire class of elements such as routers. For more information, see Section 11.2, "Determining an Element's Class Name," on page 150.
- **Class name expression:** A combination of class name and a regular expression used to further define the name of the class.
- **Script:** A script to run. Specify the script fragment, or call a script use a load statement to run from the script library. The script must evaluate to True or False. For more information about the Script Library, see the *Operations Center 5.5 Scripting Guide*.

Permission: The access privilege for the menu option: *View*, *Access*, *Manage*, or *Define*.

Type: Select one of the following actions to perform when the new menu option is selected:

- **Client script:** Runs the specified script only on the client side.
- **Server script:** Runs the specified script only on the server side.
- **Process:** Runs another program.
- **Prompt:** Identifies the prompts to display for the new menu item. Operations Center software collects this information before the operation script or process is run on the server. When the script operation is run on the server, the prompt data is supplied to the server as the `args` predefined array variable.
- **URL frame:** Applicable for CMS only.

  For more information, see "Creating Custom Operations" in the *Operations Center 5.5 Configuration Management System (CMS) Guide*.
- **URL popup:** Applicable for CMS only.

  For more information, see "Creating Custom Operations" in the *Operations Center 5.5 Configuration Management System (CMS) Guide*.
- **Web script:** Applicable for CMS only.

  For more information, see "Creating Custom Operations" in the *Operations Center 5.5 Configuration Management System (CMS) Guide*.
- **HTML result:** Applicable for CMS only.

  For more information, see "Creating Custom Operations" in the *Operations Center 5.5 Configuration Management System (CMS) Guide*.

Operation: The command line used to select the response to the new menu item. Specify a script or a command to run another process. For example, add the word `Hello` as an element menu option. When *Hello* is selected, the operation runs the "tests/hello" script from the script library on the Operations Center server. To run the *Hello* element menu option, enter:

```
operation=load("tests/hello")
```

This executes the *Hello* element menu option, as shown in the following code located in the `Operations.ini` file:

```
[Hello]
description=Say Hello to user
target=dname:
context=element
permission=view
type=script
```

In addition, you can use the operation function to launch a Web page, to show a JSP component, and so on. The */OperationsCenter_install_path*`/database/examples/Operations.ini` file contains examples on how to execute any additional commands.

4 Click *Check Syntax* to validate the syntax of the operation defined.

5 If the type selected is `Prompt`, use the following *Prompt* options to define the prompts to display before the operation executes:

Dialog Box Title: Title to display in the Prompt dialog box.

Prompt #X: Text message to show in the Prompt dialog box.

Multiline Prompt Field: Select this check box to allow the user to enter more than one line of text in response to the prompt message.

6 Click *Add* to define more than one prompt for the operation.

Additional parameters for another prompt display.

7 Click *Create*.

The new menu item displays in applicable menus.

11.4.2 Adding New Menu Items in the Operations.ini File

While this is not a recommended practice, new operations can be created by manually editing the `Operations.ini` file.

To add a new menu option using the `Operations.ini` file:

1 Do one of the following:

- Create a text file containing the new menu items.
- Copy and modify the */OperationsCenter_install_path*`/database/examples/Operations.ini` file.

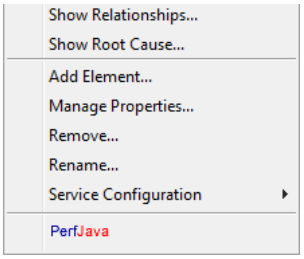
2 Save the file as `Operations.ini` in the */OperationsCenter_install_path*`/database/shadowed` directory.

To make the new menu actions accessible, you do not need to restart the Operations Center server. However, in some cases you need to log out and log back in to the Operations Center console for the changes to be visible.

11.4.3 Adding Colored Menu Operations

One way to make a custom operation stand out in the menu is to use colored text. Figure 11-3 shows the PerfJava operation in blue and red text.

Figure 11-3 *Custom Operation – Create colored operations that stand out on the menu.*



To configure colored menu options:

1. Copy and open the */OperationsCenter_install_path*/database/examples/Operations.ini file in a text editor.
2. Add the following text:

```
[PerfJava]
context=element
description=<HTML><BODY><B><FONT COLOR="Blue">Perf</FONT><FONT
COLOR="Red">Java</FONT></B></BODY></HTML>
operation=// @debug off\r\r\r\r\n\tapplet.getAppletContext().showDocument( new
java.net.URL('http://perfdata/perfjava/applet/ChartAPI.html?hostname=' +
element.name + '&title=PerfJava'), 'PerfJava')
permission=view
target=dnamematch:ms_server.*
type=clientscript
```

   Substitute the italicized text with your operation name and colors. It is assumed you understand the other options (*context*, *permission*, *target*, *type*) which are explained in Step 3 on page 152.
3. Save the file as Operations.ini in the */OperationsCenter_install_path*/database/shadowed directory.

11.4.4 Optimizing Alarm Selection

Before an operation is initiated on a set of alarms, a request is made to the server to check which operations are allowed for the alarms. This is necessary to update toolbar buttons and alarm menus. It can take a several minutes to display the menu when multiple alarms are selected.

To optimize performance:

1. Set the following parameter in the */OperationsCenter_install_path*/html/applet_params.xml file:

```
<param name="AlarmPanel.performanceToolbarMode" value="true"/>
```

   Setting the parameter to True allows the menu to display to display before the request is made to discover operations for the alarm set. However, some alarm toolbar buttons (*Close*, *Acknowledge*, and *Assign*) remain disabled until the allowed operations are returned and a menu is created.

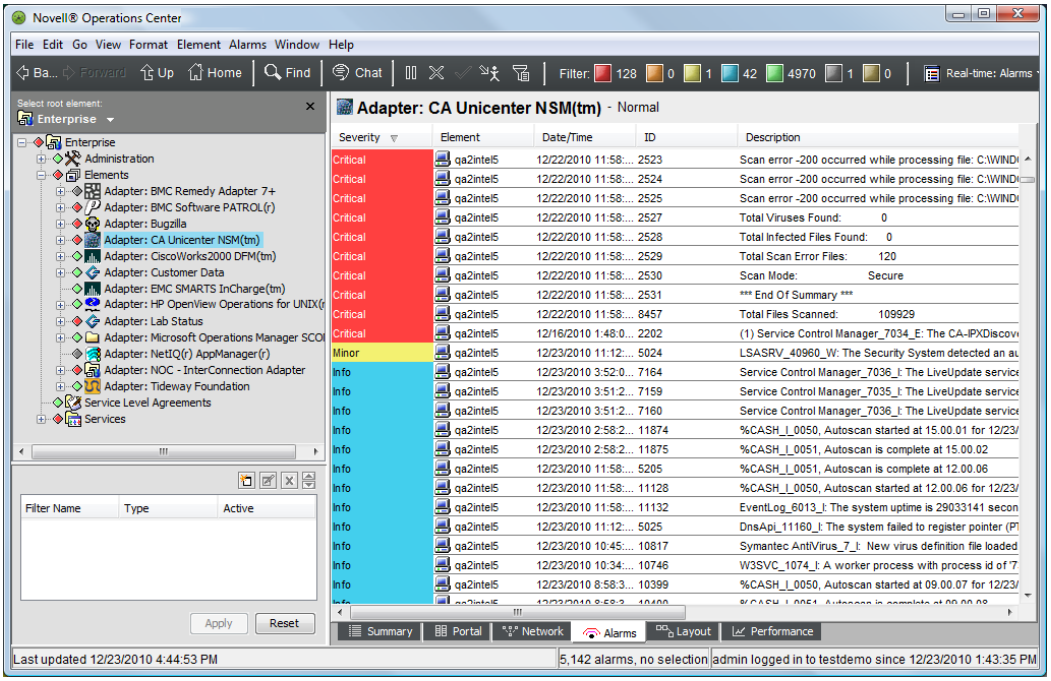
11.5 Filtering Alarm Columns

In the *Alarms* view, various columns provide information about the individual alarms:

- By default, the first four columns specify the severity, the element that generated the alarm, the date and time the alarm was reported, and a unique identification number for the alarm.
- Additional columns can vary depending on the adapter's property settings. These columns often display a text description of the alarm and other information reported by the management system.
- For adapters with an `AlarmColumns` property, the administrator can specify which columns display and the order in which they display.

  For additional information on editing adapter properties, see the *Operations Center 5.5 Adapter and Integration Guide*.

Figure 11-4 *Alarms View*



A custom property can be set to limit the number of alarms displayed as described in the following procedure.

To limit the number of alarms that display for Services and Service Models:

1 Set the following properties in the `custom.properties` file:

```
# Server.showService\ ModelsRootAlarms
#
# If set to false, alarms will not show at the Service Models root.
Server.showService\ ModelsRootAlarms=false

# Server.showServicesRootAlarms
#
# If set to false, alarms will not show at the Services Models root.
Server.showServicesRootAlarms=false
#
#If set to false, alarms will not show at the Locations root.
Server.showLocationsRootAlarms=false
```

11.6 Configuring Suppression and Acknowledgement

An element or alarm can be configured for two actions:

- Suppression
- Acknowledgement (for BMC, Tivoli, PATROL, and NetView adaptors)

Administrators can use these features to troubleshoot issues related to an alarm, or identify how a new event affects the state of an element. After a failure occurs and the root cause is determined, an operator might suppress/acknowledge the root cause element or alarm so that if a separate failure occurs that impacts the element state, the operator can quickly identify the new root cause. The actual condition of suppressed elements is still available within the server, but the parent element ignores the real state of the suppressed element when calculating its roll-up state.

Suppressing an element places the element in an unmanaged condition, which displays the element, but does not provide a condition status. A suppressed element's condition changes are ignored when calculating the roll-up state of parent elements. Suppression can be configured for a specific time interval, with an optional timeout. Suppression can be reset by a manual action, or it can expire.

Acknowledgment is available only for elements with a non-OK condition. It is not possible to schedule an acknowledgement operation to start at a future date and time. An acknowledgement can be reset by a manual action, or it can expire. It can also be reset through a manual corrective state change on one of the elements that have a non-OK state at the time of acknowledgement. Furthermore, a portion of the hierarchy that has a new failure (other than that which caused the original failure and subsequent acknowledgement) unacknowledges (turns nonbrown), but does not reset the entire acknowledgement. In this way, you can quickly determine when a new failure occurs.

Suppressing or acknowledging an element places the element in a new state. Unless the action is configured to affect only a single element, all children and their associated alarms are also suppressed or acknowledged. If configured with a time interval, these operations are reset to normal when the interval ends.

By default, users must have the Manage permission on the elements for which the Suppress or Acknowledge operations are configured in order to see and use these right-click options. However, it is possible to change the permission level required for using the *Suppress* option. For more information, see Section 11.6.2, "Changing the Permission Level for Suppression," on page 158.

To configure suppression and acknowledgement:

- Section 11.6.1, "Configuring Suppression/Acknowledgement Functions," on page 158
- Section 11.6.2, "Changing the Permission Level for Suppression," on page 158
- Section 11.6.3, "Configuring Reacknowledgement on Server Restart," on page 159
- Section 11.6.4, "Configuring Suppression/Acknowledgement to Apply to a Single Element," on page 159
- Section 11.6.5, "Enabling Suppression after Correcting Database Problems," on page 159
- Section 11.6.6, "Changing the Acknowledge and Suppress Menu Names," on page 160
- Section 11.6.7, "Using the Suppression and Acknowledgement Options," on page 160

11.6.1 Configuring Suppression/Acknowledgement Functions

Suppression and Acknowledgement functions require that the Event Data Store be configured. These functions fail unless a database definition exists for the Event Store. Also note that alarm suppression is not supported on Sybase. For more information, see Database Management Overview.

IMPORTANT: After suppression and acknowledgement are configured for one adapter, they become available system-wide for all adapters. Therefore, it is important to configure these options on an adapter that is always running, otherwise these operations become unavailable when that adapter is not running. Alternatively, set the property directly in the adapter's `adapters.ini` file, using the `Script.onStarted=@commands/suppress` command (see the next section), so that it is always available.

The suppression/acknowledgement setup script is at */OperationsCenter_install_path/database/scripts/commands/suppress.fs*. Refer to this script for instructions and examples about setting up the suppression/acknowledgement feature.

To configure suppression/acknowledgement:

1. In the *Explorer* pane, expand *Administration* > *Adapters*.
2. Right-click an adapter, then select *Properties* to open the adapter's *Status* property page.
3. In the left pane of the Properties dialog box, click *Adapter* to open its property page.
4. In the *Property* section, enter `@commands/suppress` in the *Script.onStarted* field.

   The `adapters.ini` file in the associated adapter's section now includes the following code that starts the suppression/acknowledgement:

   ```
   Script.onStarted=@commands/suppress
   ```

5. Click *Apply* to save changes, then close the dialog box.

11.6.2 Changing the Permission Level for Suppression

By default, the permission level is set to `manage`, meaning only those users who have manage permissions to an element can use the `Suppress` command.

It is possible to change the permission level for the `Suppress` command. For example, set the permission level to define to allow only those users with define privileges to an element to use the `Suppress` command.

The Tivoli T/EC adapter specifies options for suppression and acknowledgement through the AcknowledgeAvailable and SuppressAvailable adapter properties. These are both set to True by default.

To change the user permission level for the `Suppress` command, Edit the `/OperationsCenter_install_path/config/Formula.custom.properties` file and set `Suppression.SuppressPermission` to `view`, `access`, `manage`, or `define`.

11.6.3 Configuring Reacknowledgement on Server Restart

Upon restart of the Operations Center server, *Acknowledged* elements are not in an acknowledged state even if the acknowledge time interval has not expired. The `suppress.fs` script includes an option to *Reacknowledge* elements upon restart of the Operations Center server. This setting must be configured before restarting the server.

To configure reacknowledgement of elements on server restart, set the value for `state.suppress.setPersistAcks` to True in the `suppress.fs` script:

```
52  //when the Suppress subsystem was shut down.
53  state.suppress.setPersistAcks(true);
```

11.6.4 Configuring Suppression/Acknowledgement to Apply to a Single Element

Suppression applies to the entire hierarchy by default. However, you can set a custom option to limit suppression only to the selected element (does not apply to the element's children and associated alarms).

To configure suppression to apply only to the selected element, add the following line to the `/OperationsCenter_install_path/html/applet_params.xml` file between the `<common></common>` tags:

```
<param name="suppress.elementonly.checkbox" value="true" />
```

11.6.5 Enabling Suppression after Correcting Database Problems

The suppression subsystem cannot start if the Event Data Store database is missing, unavailable, or misconfigured. The following error message appears in the formula.log if the database is missing, unavailable, or misconfigured:

```
ERROR Suppression – Database definition for 'Event Store' schema is not defined or
not enabled.
```

After ensuring the database is available and correctly configured, suppression can be enabled by either restarting the Operations Center server or configuring server operations to start and stop the suppression subsystem while the server is still running.

To configure operations to enable/disable suppression:

1. In a text editor, open the */OperationsCenter_install_path*/database/shadowed/Operations.ini file and add the following two operation definitions:

```
Name: Start Suppression
Menu Text: Start Suppression
Context: Element
Match by: Distinguished name expression: ^formulaServer=Server.*
Permission: Define
Type: Server script
Operation: @commands/suppress

Name: Stop Suppression
Menu Text: Stop Suppression
Context: Element
Match by: Distinguished name expression: ^formulaServer=Server.*
Permission: Define
Type: Server script
Operation: @commands/suppressoff
```

   Adding a new operation does not require the Operations Center server to be restarted, but might require you to log out and back in again.

   For instructions about creating operation definitions, see Section 11.4, "Modifying Element and Alarm Menus," on page 151.

2. Right-click the *Server* element, then select *Stop Suppression*.
3. Right-click the *Server* element, then select *Start Suppression* to restart the suppression subsystem.

11.6.6 Changing the Acknowledge and Suppress Menu Names

To change the default *Suppress* and *Acknowledge* menu option names:

1. In a text editor, open the */OperationsCenter_install_path*/config/Formula.custom.properties file.

   Use a Formula.custom.properties file so that Configuration Manager does not overwrite customized properties.

2. Enter the following lines into the file replacing new name with the new menu option for the properties.

```
operation.suppress.label=new name
operation.unsuppress.label=new name
operation.acknowledge.label=new name
operation.unacknowledge.label=new name
```

3. Save the file.
4. Restart the Operations Center server.

11.6.7 Using the Suppression and Acknowledgement Options

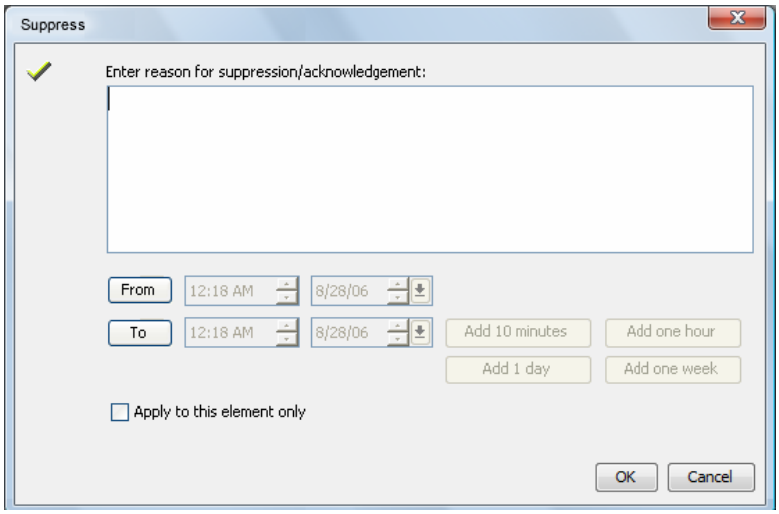
After configuring the suppression and acknowledgement functionality, the *Suppress* and *Acknowledge* options are available for use.

- "Suppressing or Unsuppressing an Element or Alarm" on page 161
- "Displaying the Reason for a Suppressed Element" on page 162
- "Unsuppressing an Element" on page 162
- "Acknowledging an Element" on page 163

Suppressing or Unsuppressing an Element or Alarm

To suppress/unsuppress an element or alarm:

1. In the *Explorer* pane, expand *Elements*.
2. Right-click the element, then select *Suppress* to open its dialog box.
3. Specify the reason for suppressing the element or alarm.



The *OK* button is not activated until you enter a reason for suppression.

To leave the reason blank, enter one character, then backspace to delete it.

If no reason is provided, the element displays the reason provided for the first parent element in the hierarchy that can provide a suppression reason.

4. (Optional) Use the *From* and *To* spinners to define the starting and ending date and time interval during which the element or alarm is suppressed.
5. (Optional) Click *Add 10 Minutes*, *Add One Hour*, *Add 1 Day*, or *Add One Week* to increase the time during which the suppressed state is applicable.
6. Select the *Apply to This Element Only* check box to specify that the suppression does not apply to any children or their associated alarms.

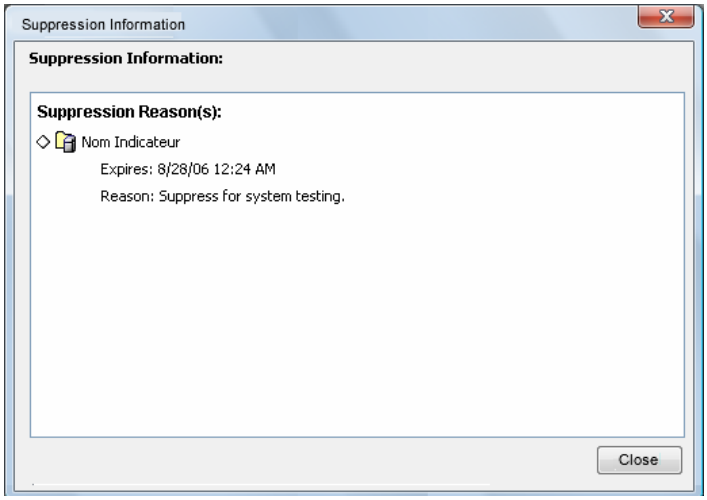
   If not selected, suppression affects child elements and their alarms.
7. Click *OK*.

Displaying the Reason for a Suppressed Element

To view the reason for an element suppression:

1 Right-click the suppressed element, then select *Show Reason*.

The Suppression Information dialog box opens and displays the reason the element was suppressed:



2 Place the mouse over the suppressed element to display a tool tip that shows the location of the element in the hierarchy.

Unsuppressing an Element

To remove the suppression for an element:

1 Right-click a suppressed element, then select *Unsuppress*.

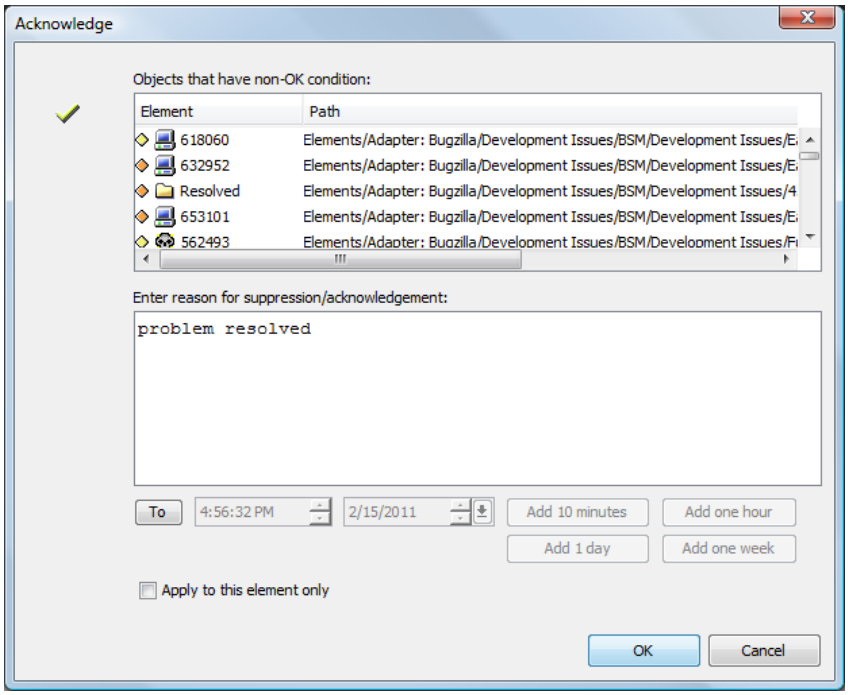
The element is no longer suppressed.

Acknowledging an Element

To acknowledge an element:

1. In the *Explorer* pane, expand *Elements*.
2. Navigate to the element to be acknowledged, right-click the element, then select *Acknowledge*.

   The Acknowledge dialog box opens and displays a list of elements with a non-OK condition:



3. Specify the reason for acknowledging the element or alarm.
4. (Optional) Use the *To* spinner to set the time and date when the acknowledgement is no longer applicable and expires.
5. (Optional) Click *Add 10 Minutes*, *Add One Hour*, *Add 1 Day*, or *Add One Week* to increase the time during which the acknowledged state is applicable.
6. Select the *Apply to This Element Only* check box to specify that the acknowledgement does not apply to any children or their associated alarms.

   If not selected, acknowledgement affects child elements and their alarms.
7. Click *OK*.

11.7 Controlling Display of Alarms

- Section 11.7.1, “Configuring Maximum Alarms,” on page 164
- Section 11.7.2, “Limiting the Roll Up of Alarms to Root Elements,” on page 164
- Section 11.7.3, “Displaying a Message for Paused Alarms,” on page 165
- Section 11.7.4, “Resuming Alarms after Pause,” on page 165
- Section 11.7.5, “Pause Alarms on Right-Click,” on page 165

11.7.1 Configuring Maximum Alarms

For performance reasons, Operations Center software is set by default to restrict the maximum number of alarms displayed in the Operations Center Console and the dashboard.

The Operations Center Console does not send requests to the Operations Center server for alarms of any nodes whose alarm flags are set to False.

To change the default setting for alarms limits:

1. Add the `HyperQuery.NumPoints.SizeLimit` property to the `/`*`OperationsCenter_install_path`*`/config/Formula.custom.properties` file.

   For example, the following code sample illustrates the entry to limit the number of alarms to 1000:

   ```
   HyperQuery.NumPoints.SizeLimit=1000
   ```

2. Start/Restart the Operations Center server.

11.7.2 Limiting the Roll Up of Alarms to Root Elements

For performance reasons, Operations Center software is set to restrict the roll up of alarms to the *Enterprise*, *Elements*, *Locations*, *Services*, and *Service Models* root elements. Significant performance impact can occur if users attempt to view all alarms for the entire Operations Center server (such as at the Enterprise root). The roll up of alarms is allowed to the *Administration* root element, which is useful when auditing is enabled.

To change the default system behavior for alarms roll-up:

1. Stop the Operations Center server.
2. Update the `/`*`OperationsCenter_install_path`*`/config/Formula.custom.properties` file.

   For example, the following code samples illustrate the entries for the *Enterprise* and *Elements* root nodes, which set the roll up of alarms to those elements:

   ```
   # If set to false, alarms will not show at the enterprise level.
   Server.showEnterpriseAlarms=true

   # If set to false, alarms will not show at the Elements root.
   Server.showElementsRootAlarms=true
   ```

   When flags are set to True, users can view all alarms under the root element. However, setting these flags to True can significantly impact performance when retrieving every alarm for the entire Operations Center system (for example, at the Enterprise root).

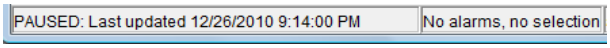
   If any of the showRootElementAlarms flags are set to False, all users are denied the ability to view a roll up of alarms at the corresponding root element. This includes new real time alarms, as well as historical alarms. The *Alarms* view can be selected, but the data display is empty.

3. Start the Operations Center server.

11.7.3 Displaying a Message for Paused Alarms

In the *Alarms* view, when the user selects an alarm, Operations Center software dynamically pauses all updates. The status bar in the bottom left of the Operations Center console displays a PAUSED message with the corresponding date and time:

Figure 11-5 *Alarm Status Bar*



Also, the *Resume* button on the *Alarms* view toolbar blinks when alarms are paused.

To emphasize that the *Alarms* view is paused, use a script to display a pop-up alert when the *Alarms* view is in Pause mode for a specified amount of time.

Displaying the alert requires using two scripts from the script repository:

`/OperationsCenter_install_path/database/scripts/util/pauseStop_op.fs`

`/OperationsCenter_install_path/database/scripts/util/pause_op.fs`

For more information about using these scripts, refer to instructions at the top of the `pause_op.fs` file.

11.7.4 Resuming Alarms after Pause

Custom settings can be made that control the Resume behavior in the *Alarms* view.

To customize the Resume behavior for alarms:

1. Edit the `/OperationsCenter_install_path/html/client/template/launch.jnlp` file, then add the following property assignment in the `resources` section:

   ```
   <property name="Client.alarm.resume.delay" value="-1/">
   ```

   Do one of the following:

   - Set the value to -1 to pause alarms until the *Resume* toolbar button is clicked.
   - Set the value to 0 to resume alarms when the *Alarms* view closes.
   - Set the value to any number greater than one (1, 2, …N) to resume alarms after the specified number of seconds.

2. Run the Operations Center Configuration Manager and click *Apply*.

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

11.7.5 Pause Alarms on Right-Click

Custom settings can be made to pause the Alarms View and halt the visual update of alarm data when using right-click operations. This ensures that a right-click operation does not lose it's context if new alarm data is being received.

To halt alarm refresh for alarms on right-click:

1. Edit the `/OperationsCenter_install_path/html/client/template/launch.jnlp` file, then add the following property assignment in the `resources` section:

   ```
   <property name="Client.alarm.pause.right.click" value="true"/>
   ```

Alarms are paused for right-click operations. For information on setting `Client.alarm.resume.delay` property to resume alarms after a specified time, see Section 11.7.4, “Resuming Alarms after Pause,” on page 165.

2 Run the Operations Center Configuration Manager and click *Apply*.

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

11.8 Filtering Service Model Alarms

Administrators can create alarm filters and filter groups and apply them to service models or metamodel classes. Instance-level filters always take priority over class-level filters. These filters determine the subset of alarms that users can view in the Operations Center console and Dashboard.

IMPORTANT: Alarm filtering occurs on the client side. Any server side scripts that perform the `getAlarms` function on an element with alarm filters returns all alarms, unfiltered.

The following sections explain how to use filtering for Service Model alarms:

- Section 11.8.1, “Creating Alarm Filters,” on page 167
- Section 11.8.2, “Editing Alarm Filters,” on page 169
- Section 11.8.3, “Creating Filter Groups,” on page 170
- Section 11.8.4, “Applying Filters to Service Models,” on page 171
- Section 11.8.5, “Applying Filters to Element Classes,” on page 172
- Section 11.8.6, “Clearing Filters from Elements,” on page 173
- Section 11.8.7, “Giving Users View Permissions to Filters,” on page 173
- Section 11.8.8, “Deleting Filters,” on page 173
- Section 11.8.9, “Assigning Filters Using Scripts,” on page 174

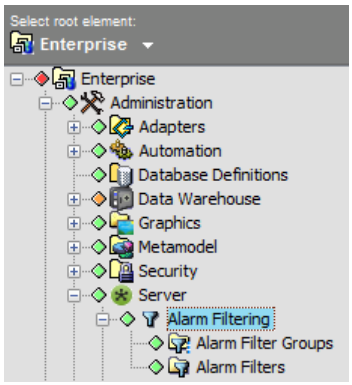
11.8.1 Creating Alarm Filters

Filters for service model alarms are created independently of an element and are then applied to service models. Because service model alarm filters are not attached to specific element on creation, a sampling of alarm columns from a non-related element are used to build the filter before it is applied.

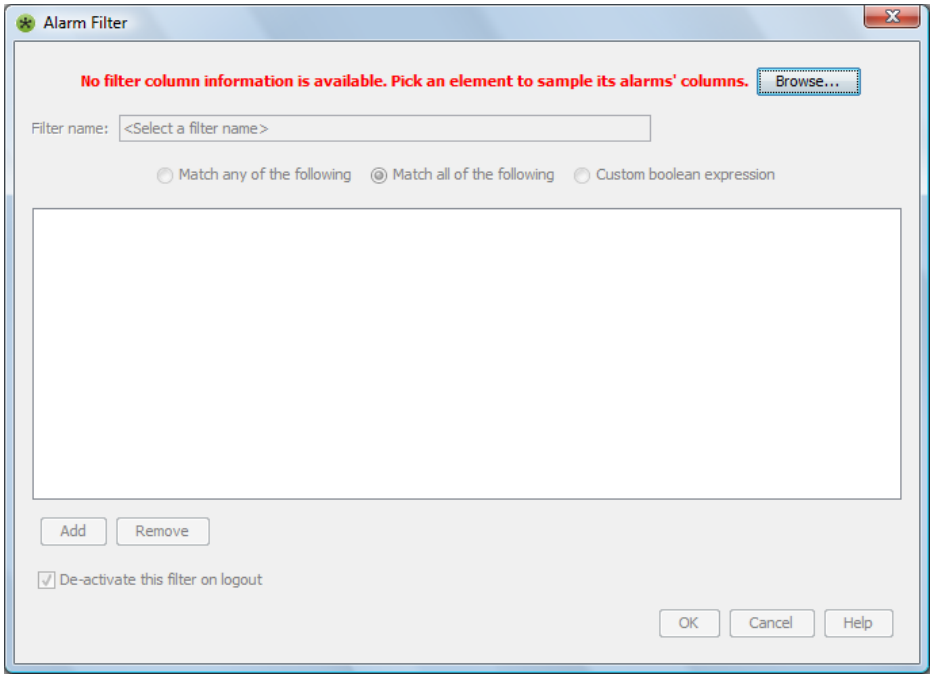
The rules used for alarm filters that are applied to service models and metamodel classes are the same as those described earlier in the section. For details on the appropriate usage and syntax of conditional statements in a filter, see "Creating an Alarm Filter" in the *Operations Center 5.5 User Guide* and see the table in Step 8 of the "Creating an Alarm Filter" procedure.

To create an alarm filter:

1 In the *Explorer* pane, expand *Administration* > *Server* > *Alarm Filtering*.



2 Right-click *Alarm Filters*, then select *Create Alarm Filter*. The *Alarm Filter* dialog box opens.



3 Click *Browse* to select an element that currently has alarms.

This is used to gather a sample list of alarm column names which can prepopulate the alarm column field for defining the filter. We recommend selecting an element that has the widest wide-range of alarm columns.

4 Select the element, then click *OK*.

Selectors in the Alarm Filter dialog box activate.

Operations Center determines the alarm column names and types are available based on the alarms of the selected element.

5 Specify a name for the new filter in the *Filter Name* field.

6 Do one of the following to determine the type of match: Select the *Match All* radio button to determine if an alarm must match any or all of the condition statements in order to be selected.

- Select the *Match any of the following* radio button to match one or more condition expressions. Selecting this option joins more than one statement with an OR operator.
- Select the *Match all of the following* radio button to match all of the condition expressions. Selecting this option joins more than one statement with an AND operator.
- Select the *Custom boolean expression* radio button to join the statements using a custom boolean expression.

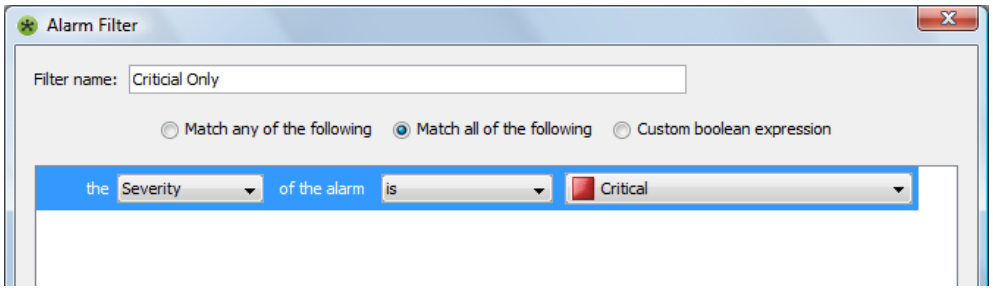
7 Click *Add*.

A blank row displays for defining a filter statement.

8 Click the first drop-down list, then select the alarm column name to match in the filter.

The list options available depend on the selected element in see the table in Step 8 of the "Creating an Alarm Filter" procedure in the *Operations Center 5.5 User Guide*.

For example, select *Date/Time* if the filter searches alarms based on the date and time they were received:



9 Select comparison operators and specify match criteria using the other fields in the row. The list or options vary depending on the type of alarm column selected from the first drop-down list.

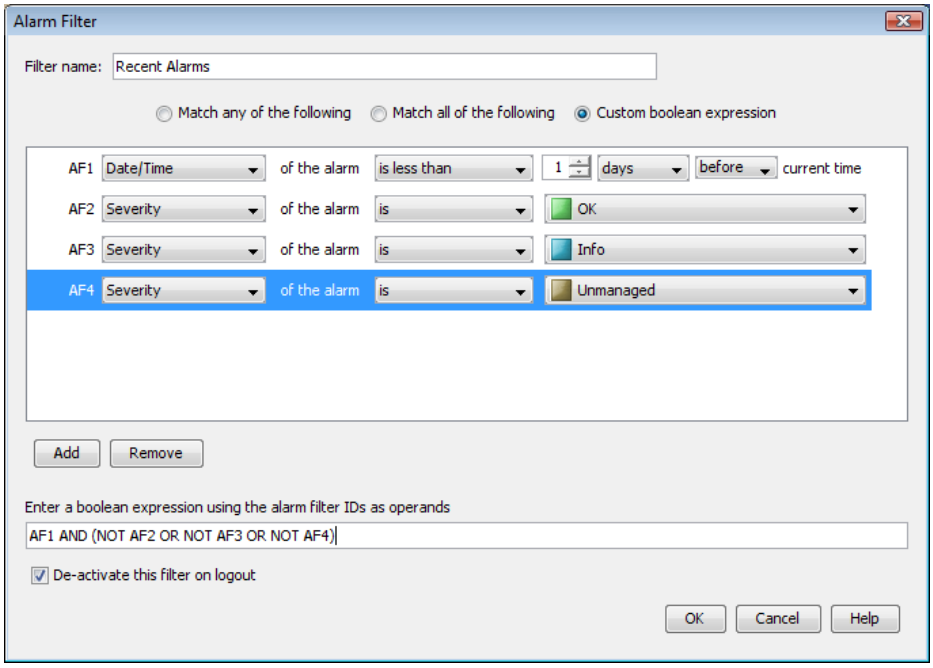
For more information, see the table in Step 8 of the "Creating an Alarm Filter" procedure in the *Operations Center 5.5 User Guide*.

10 To add an additional filter expression, click *Add*.

11 If the *Custom boolean expression* radio button was selected in Step 6, specify a boolean expression using alarm expression ids to indicate how the condition statements are to be applied:

- Expression ids display in the front of each condition statement.
- Use AND to join two expressions.
- Use OR to match any of two expressions.
- Use NOT to not match an expression.

- The boolean expression is not case-sensitive. Not all alarm expressions must be applied as boolean expressions can be formed in a way that ignores an expression. For example, if the following expression is used, `(AF1 OR AF2) AND (NOT AF4)`, then AF3 would be ignored.
- In the figure below, the filter allows alarms no older than 24 hours with a severity of CRITICAL, MAJOR, MINOR and UNKNOWN.



12 (Conditional) To deactivate the filter upon logout from the Operation Center console, select the *Deactivate this filter on logout* check box. This option only applies when filter is applied to non-Service Model elements.

To leave the filter activated from session to session, leave the check box deselected.

Deactivating filters upon logout ensures that all alarms display after the next login.

If filters remain active upon logout, the active filters apply upon login and only the selected alarms display in the *Alarms* view. Problems might arise from not realizing that all alarms are not displayed.

13 Click *OK* when all filter criteria has been defined.

11.8.2 Editing Alarm Filters

Because service model alarm filters are not attached to specific element on creation, it is necessary to obtain another sampling of alarm columns to prepopulate fields when you are editing alarm filters.

To edit an alarm filter:

1 In the *Explorer* pane, expand *Administration* > *Server* > *Alarm Filtering* > *Alarm Filters*.

2 Right-click the desired filter, then select *Properties*. The *Properties* dialog opens.

3 Click *Alarm Filter* to open the filter definition.

4 To edit the alarm column for the existing filter expressions or if planning to add additional expressions, do the following:

4a Click *Sample* to select an element that currently has alarms.

This is used to gather a list of alarm column names which prepopulate the alarm column field for defining the filter. We recommend selecting an element under Service Models that has a wide-range of alarm columns.

4b Select the element, then click *OK*.

Selectors in the Alarm Filter dialog box activate.

Operations Center determines the alarm column names and types are available based on the alarms of the selected element.

5 To add an additional filter expression, click *Add*.

For details on adding filters as well as the appropriate usage and syntax of conditional statements in a filter, see "Creating an Alarm Filter" in the *Operations Center 5.5 User Guide*.

6 Click *OK*.

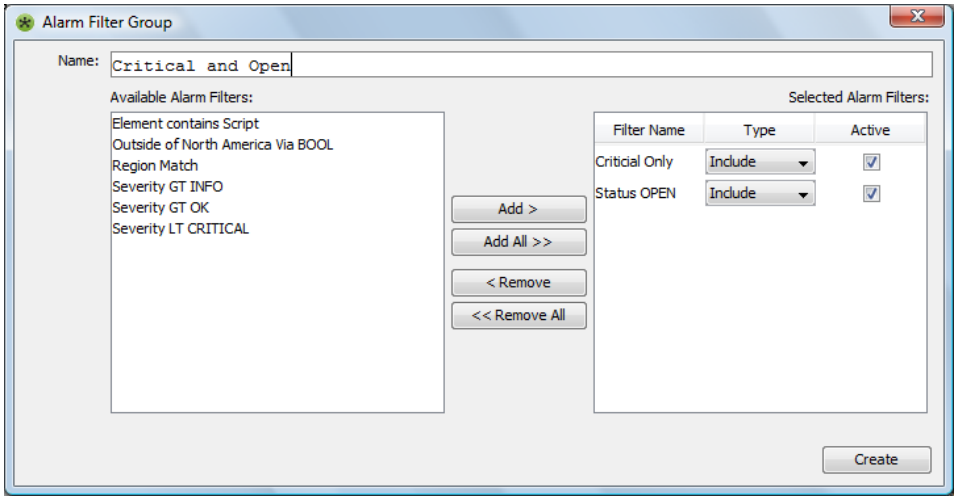
11.8.3 Creating Filter Groups

Use alarm filter groups to apply multiple filters to a service model. Note that all filters in a group are applied using the AND operator, meaning an alarm must meet the conditions stated by all filters in the group in order to be displayed in the *Alarms* view.

To create an alarm filter group:

1 In the *Explorer* pane, expand *Administration* > *Server* > *Alarm Filtering*.

2 Right-click *Alarm Filter Groups*, then select *Create Alarm Filter Group* to open the Alarm Filter Group dialog box:



3 In the *Name* field, specify a name for the new group.

4 Select one or more alarm filters in the *Available Alarm Filters* field, then click *Add* to add the filters to the group.

The order of filters is significant, as filters are processed from the top down. Only the alarms that pass the first filter are tested by the second filter, and so on.

The selected filters move to the *Selected Alarm Filters* field.

5 For each selected filter, decide to include or exclude matching alarms from the *Alarms* view:

- To show alarms selected by the filter, click the associated *Type* drop-down list, then select *Include*.
- To hide alarms selected by the filter, click the associated *Type* drop-down list, then select *Exclude*.

  Alarms selected by the filter are not displayed in the *Alarms* view.

6 If there is a need to stop using a filter temporarily, unmark the *Active* check box.

  By default, defined filters are active, meaning they are applied to the *Alarms* view.

7 When finished, click *Create*.

11.8.4 Applying Filters to Service Models

Alarm filters can be applied at the class-level, or with an element-level filter or filter group. Instance-level filters always take priority over class-level filters. When you create a class-level filter, it applies to all elements in the class, except the elements that have their own filters/ filter groups. Class-level filters are ignored by these elements.

You can ignore the class-level filter even when there is no element-level filter. Open the element's *Alarm Filtering* property page, then select *Ignore Class-Level Filters*. For more information, see Section 11.8.5, "Applying Filters to Element Classes," on page 172.

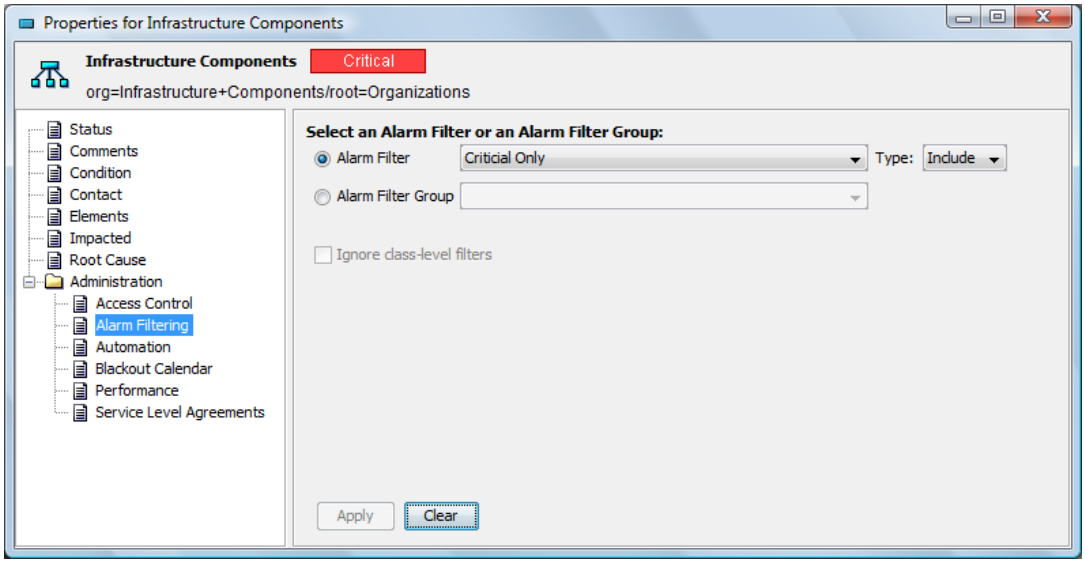
Alarm filters and filter groups can be applied to any element in the *Service Model* hierarchy.

To apply server-side filters to elements under *Service Models*:

1 Right-click an element in the *Service Models* hierarchy, then select *Properties*.

2 In the left pane of the Properties window, click *Alarm Filtering*.

3 Select either the *Alarm Filter* or *Alarm Filter Group* radio button:



4 Select the filter or filter group from the drop-down list.

  If you select a filter, decide whether to include or exclude alarms that are collected by the filter. Include displays the alarms in the *Alarms* view; exclude prevents them from displaying.

The *Ignore Class-Level Filters* option is relevant if alarm filters are applied to the element class (see the next section).

This option is available only if there are no alarm filters or groups applied to the element on the *Alarm Filtering* property page shown above. Otherwise, the option is dimmed and cannot be selected. As stated earlier, when an element-level filter or filter group is applied, the class-level filter is ignored . If you do not want to apply the class filter to a specific element, select the *Ignore Class-Level Filters* check box.

If you have already made a filter or filter group selection on the *Alarm Filtering* property page, click *Clear* and then you can select the *Ignore Class-Level Filters* check box.

5 Click *Apply*.

11.8.5 Applying Filters to Element Classes

Alarm filters and filter groups can be applied to a class of elements. The filters are applied to all elements of the specified class. The exceptions are:

- Ignore class-level filters is selected on an element's *Alarm Filters* property page. For more information, see the previous section.
- A filter (or filter group) is applied to an element. This filter takes priority over the class-level filter, which is ignored.

To apply a filter to a class:

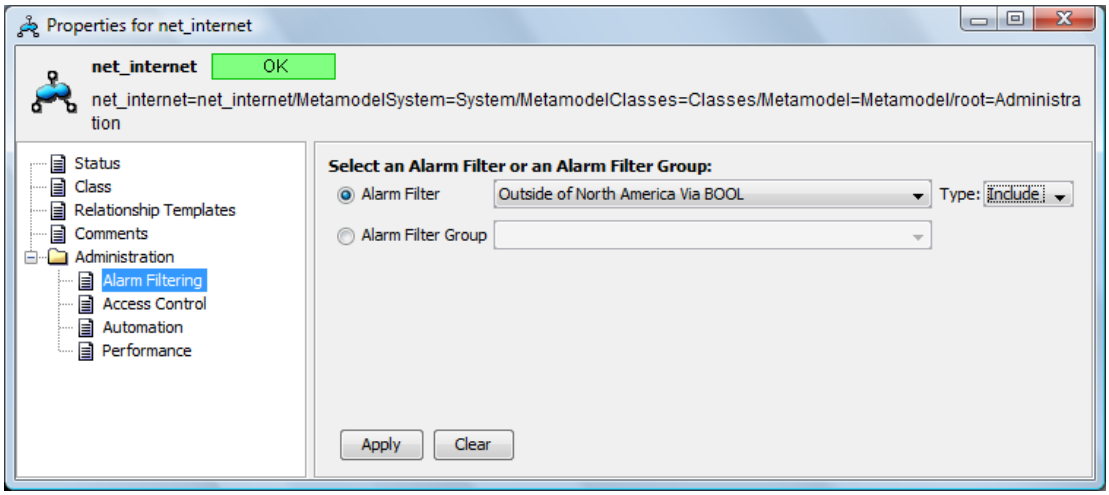
1 In the *Explorer* pane, expand *Administration* > *Metamodel* > *Classes*.

2 Right-click a class, then select *Properties*.

3 In the left pane of the Properties window, click *Alarm Filtering*.

4 Select either the *Alarm Filter* or *Alarm Filter Group* radio button.

5 Select the filter or filter group from the drop-down list:



6 Decide whether to include or exclude alarms that are collected by the filter.

Included alarms display the alarms in the *Alarms* view; excluded alarms do not display.

7 Click *Apply*.

11.8.6 Clearing Filters from Elements

To clear a filter, open the *Alarm Filtering* property page for the *Service Model* element or class, then click *Clear*.

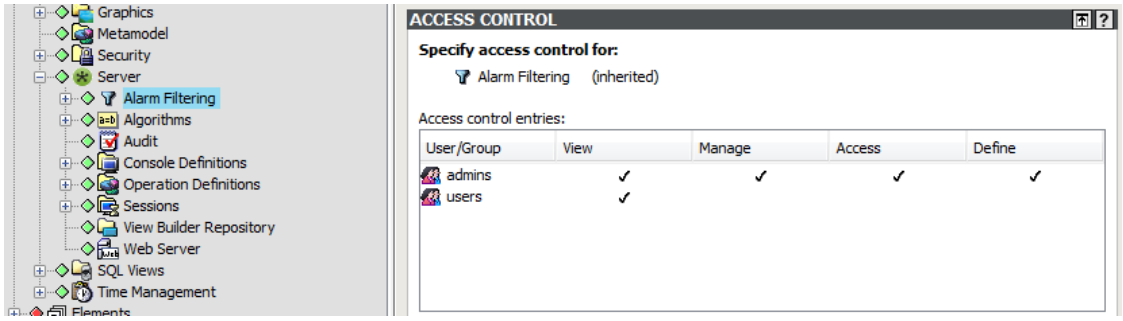
11.8.7 Giving Users View Permissions to Filters

Users must have a minimum of View permissions to the *Alarm Filters* and *Alarm Filter Groups* element in order for filtering to work. If a user does not have View permission, then the client session cannot obtain the filter definition from the server, and the user can view all unfiltered alarms.

The default Users group does have View permissions. You should consider granting View permissions to other affected user groups.

To view or edit group permissions:

1. Select the *Alarm Filters* or *Alarm Filter Groups* element under *Administration* > *Server* > *Alarm Filtering*.
2. Open the *Portal* view:



Group permissions are listed in the *Access Control* pane.

11.8.8 Deleting Filters

When an alarm filter is deleted, the filter remains assigned to the service model. The service model is not scanned to remove any assignment of this alarm filter/filter group because of potential performance degradation. However, deleted filters are ignored during alarm processing.

These stale filter assignments are removed the next time the *Alarm Filtering* property page is opened. The element is updated and the deleted filter is removed. If a class-level filter exists, it is applied then.

To delete a filter or filter group:

1. In the *Explorer* pane, expand *Administration* > *Server* > *Alarm Filtering* > *Alarm Filters* or *Alarm Filter Groups*.
2. Right-click a filter or filter group, then select *Delete Alarm Filter* or *Delete Alarm Filter Group*.

11.8.9 Assigning Filters Using Scripts

In many situations, it more convenient to use scripts to assign alarm filters and filter groups. Examples include during SCM build jobs and environments where administration is automated through scripts.

To assign alarm filters and filter groups programmatically:

1 In the script, use the following syntax:

```
element._mosolSmaf = 'smafName'
```

where the format of the *smafName* is one of the following:

- AFG:*AFG_name*

  Assigns an existing alarm filter group to the service model.

- AFI:*AF_name*

  Assigns an existing alarm filter and includes the filter results,

- AFE:*AF_name*

  Assigns an existing alarm filter and excludes the filter results,

The alarm filter and alarm filter group names must already exist in the system. They do not need to be URL encoded.

Some examples:

- Assume there is an alarm filter group named `critical db apps` and an alarm filter named `hosts in domain X`. The following function assigns the `critical db apps` alarm filter group to the element:

  ```
  element._mosolSmaf = 'AFG:critical db apps'
  ```

- The following function assigns the `hosts in domain X` alarm filter to the element. The alarm filter is the Include type, meaning that alarms selected by the filter are displayed in the *Alarms* view:

  ```
  element._mosolSmaf = 'AFI:hosts in domain X'
  ```

  The above code assumes that the variable `element` is in scope.

  If you were to enter `AFG:hosts in domain X` in the script, it is an invalid value because there is a filter, but not a filter group named `hosts in domain X`. Similarly, `AF:critical db apps` is invalid because the AF prefix is invalid; only the AFG, AFI, and AFE prefixes are valid.

11.9 Managing Administration Elements on Remote Servers

It is possible to view and manage the *Administration* element branch of a remote server. You can use the InterCommunication adapter (ICA) to establish communication between the two machines.

The following functionality is unavailable on remote servers that are accessed using the ICA:

- *Session* element:
  - *Chat* operation
  - *Chat* button on the *Session* property page

- *BDI* element:
  - `Import Definition` command
  - `Import` command in the *BDI File* menu
- `View Builder Element Export` command
- *Profiles* element:
  - Create Profile operation
  - *Profile* property page
  - *Elements* property page
- *Automation Editor* property pages for *Automation* elements
- *Adapter* property page real-time status updates, including the *Start* and *Stop* buttons
- *Calendar* element:
  - *Create Calendar* operation displays as read-only
  - *Edit Calendar* operation displays as read-only

To configure ICA:

1. When defining adapter properties for the ICA, set `Discover Administration` to `True`.

   This enables viewing and performing some management tasks for the remote Administration branch. If `Discover Administration` is set to False, you cannot view the remote branch.

   Defining an adapter and its properties is explained in the *Operations Center 5.5 Adapter and Integration Guide*.
2. Because it is possible using the ICA to monitor remote server and warehouse element performance in real-time and also using SLM, view the properties of these remote elements and optionally add them to local performance and/or SLA profiles.
3. Perform any of the following administration tasks using the ICA:
   - Create, edit, start, and stop adapters remotely
   - Create and edit operations remotely
   - Create and edit automations remotely
   - Set audit alarms so they pass from the ICA server to the local server under the Administration branch

12 Capturing Alarm and Performance History

Operations Center provides features to capture, store, and display alarm history and historical performance information. This section explains the setup requirements to store alarm history and historical performance, as well as how to create schedules and specify which data to collect.

For more information about viewing and filtering alarm history information and charting performance data in the console, see "Viewing and Managing the Alarm History" and "Charting Performance Data" in the *Operations Center 5.5 User Guide*.

Alarm history data can be captured and viewed with a standard Operations Center license. However, capturing and viewing performance and Service Level metrics (SLAs) information requires purchasing a Operations Center SLM license.

- Section 12.1, "Alarm History Overview," on page 177
- Section 12.2, "Using Profiles and Expressions to Capture Alarm History," on page 178
- Section 12.3, "Assigning Profiles at the Element Level," on page 192
- Section 12.4, "Viewing Alarm History," on page 192
- Section 12.5, "Monitoring Alarm History Collection," on page 193
- Section 12.6, "Customizing Default Performance Chart Settings," on page 195

12.1 Alarm History Overview

The requirements for setting up a system to store alarm history include a separate database definition and calendars and schedules that define when to capture historical data. Profiles are required to define exactly what data are captured.

A database definition establishes the configuration settings and other parameters required to establish a connection between Operations Center and a specific database.

The general steps to store alarm history:

1. Configure and enable a database definition to connect with your database.

   For more information, see Section 7.5, "Configuring the Service Warehouse," on page 90.

2. Edit time categories as required and create one or more linked calendars and schedules for capturing and storing data.

   For more information, see Chapter 13, "Time Categories, Calendars, and Schedules," on page 197.

3. Set up and activate a profile that contains an Alarm History expression.

   The profile and expression determine which specific properties are stored when the profile runs. The profile is linked to a schedule, which determines when the alarm data is captured.

After capturing and storing alarm history, do one of the following to view the alarm history:

- View alarm history for elements by using the *Historical: Alarms* option in the *Alarms* view.

  Change the alarm display using the options listed under the *Format* menu. These options include defining the maximum number of historical alarms displayed.
- Double-click an alarm recorded, then select the *History* tab to view the history of a single alarm.

To store alarm history, SLA data, or performance data, the element properties created by an adapter must meet the supported schema as specified in the Data Dictionary. Otherwise, a "value too large" error occurs, displaying the maximum allowed length for a property value and the actual length. For example, DNames cannot exceed 3,000 characters.

12.2 Using Profiles and Expressions to Capture Alarm History

The Data Warehouse uses profiles to select the elements for which alarm history is collected. The matching criteria can be based on specific elements, element classes, regular expressions, or scripts. Attach a schedule to a profile to define the dates and times for collecting alarms history.

Profiles can also capture historical performance data used in the *Performance* view, which displays performance data for one element at a time, and for the Performance Analysis dialog box, which displays performance data for multiple elements at the same time.

An Alarm History expression is defined for a profile, instructing how to save alarms in the historical database. With regard to service level management, expressions determine which properties of an element are stored.

- Section 12.2.1, "Creating Profiles," on page 178
- Section 12.2.2, "Creating Expressions," on page 184
- Section 12.2.3, "Starting and Stopping Profiles," on page 189
- Section 12.2.4, "Monitoring Profile Data Collection," on page 191

12.2.1 Creating Profiles

Profiles are used to select the elements for which performance and alarm data is collected. Element selection is based on user-defined match criteria. After the elements are selected, expressions are added to the profile to determine which element properties are stored. A schedule is attached to the profile to determine when the data is collected. The profile has retention settings to determine when the data is purged. There is also a setting for when the profile starts.

Match criteria for the profile can be based on specific elements, element classes, regular expressions, or scripts.

Data collected for a profile is retained on a rolling basis. Specify the number of days by entering a value for the *Retain This Data for X Days Option* when creating a profile. For example, if the value entered is 30, historical data is retained for the past 30 days. Older data is discarded. Data is discarded as part of the historical data purge job. When the data retention setting is changed for a profile, the new retention time applies only to data collected from that point forward.

- "Understanding Default Service Level Management Profiles" on page 179
- "Creating a Profile" on page 180
- "Using Profile Match Criteria to Select Objects" on page 181

- "Editing Profiles" on page 183
- "Deleting a Profile" on page 184

Understanding Default Service Level Management Profiles

Operations Center ships with two default profiles for capturing service level management data:

- Element Condition profile stores real-time element condition data
- Service Levels profile exists for automatic storage of service level metrics

Both profiles contain predefined Historical Performance expressions. The data retention settings for these profiles can be modified, but the profiles themselves cannot be edited or deleted.

The Service Levels profile has one predefined expression named Element Condition Change. This expression captures real-time condition changes including service level breach and warning alarms, outages, root cause data, and any other data required to support service level monitoring and management. This data is recorded in the Service Warehouse regardless of Service Warehouse settings.

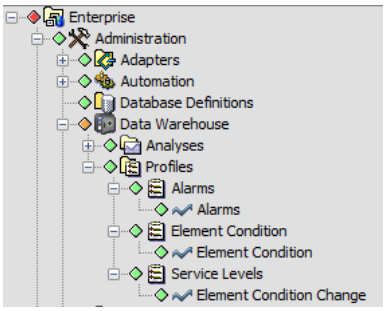
There is no need to add expressions to the Service Levels profile, but you have the ability to do so. However, carefully consider the number of expressions that you add, as the amount of data stored can increase significantly.

The Service Levels profile captures data for all elements with Service Level Agreements Service Level profile information. For this reason, the Matches function is disabled. Instead, elements are added in the background when a Service Level Agreement is applied to them.

The schedule for the Service Levels profile is also permanently set to one minute. This schedule merely represents the frequency with which data is updated in the Service Warehouse. Service level data is constantly captured.

Profiles are listed in the *Explorer* pane under the *Administration* or *Data Warehouse* element. Default profiles (Service Levels, AuditProfiles) cannot be edited or deleted. However, their data retention settings can be modified.

Figure 12-1 *Explorer Pane: Profiles Are Found Under the Data Warehouse Element in the Administration Root.*

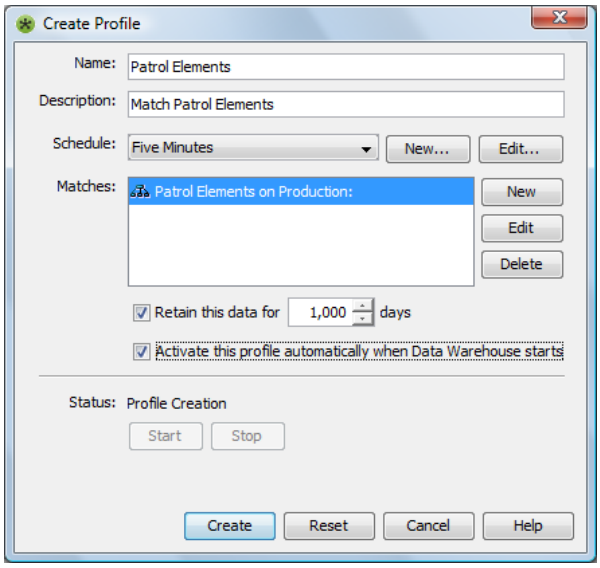


The next step to setting up the Data Warehouse to collect alarm history data is to create an Alarm History expression for the profile. For more information, see "Alarm History Expressions" on page 185.

Creating a Profile

To create a profile:

1. In the *Explorer* pane, expand *Administration* > *Data Warehouse*.
2. Right-click *Profiles*, then select *Create Profile* to open its dialog box:



3. Fill in the fields:

   Name: The profile name.

   Description: The profile's purpose.

   Schedule: Select a previously defined schedule for collecting data for this profile.

   Matches: Click the *New* button (next to the *Matches* section) to add criteria for selecting objects. Matches can be made using specific elements, element classes, regular expressions, or a script. Do the following as necessary:

   - To edit a match, select an item in the *Matches* section, then click *Edit*.
   - To remove criteria, select an item in the *Matches* section, then click *Delete*.

   For more information, see to "Using Profile Match Criteria to Select Objects" on page 181 additional information.

   Retain this data for: To retain performance data for a specific number of days, select the check box and use the spinners to set a number.

   Data is retained on a rolling basis. For example, enter 30 and historical data is retained for the past 30 days only. Older data is discarded.

   This setting works in conjunction with the BSW Historical Data Purge, which is a default job, defined under the *Administration* > *Time Management* > *Jobs* element.

   Activate this Profile when Data Warehouse starts: To start the profile when Operations Center software launches, select the check box.

   If the check box is not selected, the profile can be started manually by clicking *Start* in the Properties dialog box.

   It is necessary to define expressions before starting a profile.

4 Do one of the following to complete the process of creating the profile:

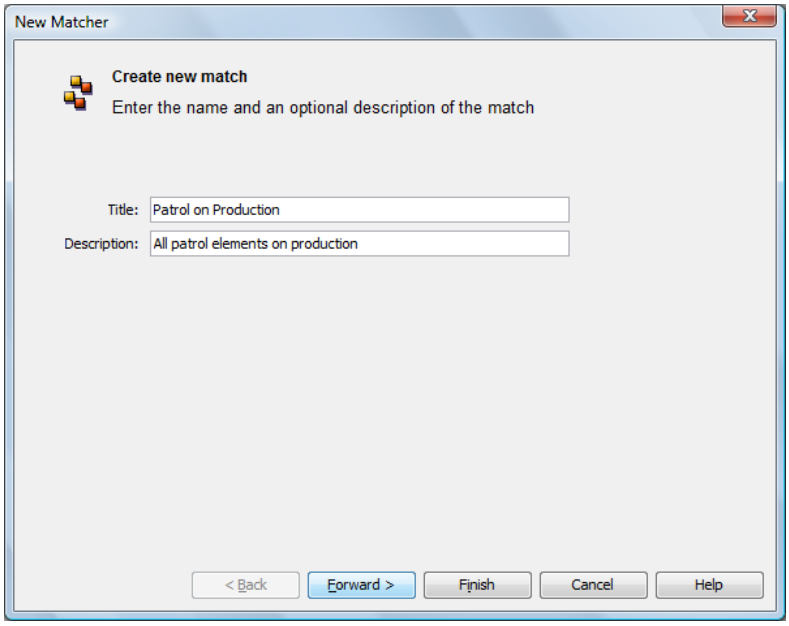
- Click *Create* to create the profile.
- To abandon creating the profile and close the dialog box, click *Cancel*.

Using Profile Match Criteria to Select Objects

A profile can select objects by specific element, class name, regular expression, or script-based matching.

To specify profile matching criteria:

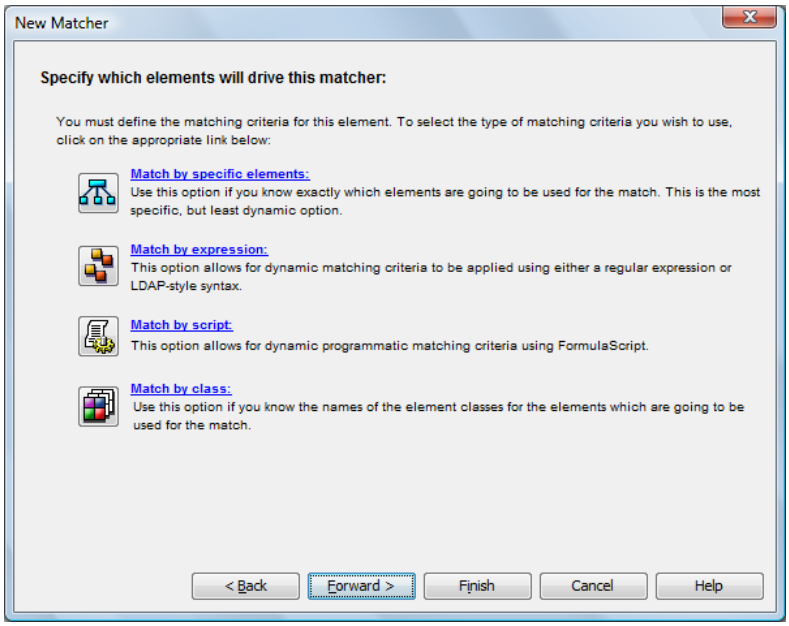
1 From the Create Profile dialog box, click the *New* button (next to the *Matches* section) to open the New Matcher wizard:



2 Specify a name for the profile in the *Title* field.

3 Specify a description for the profile in the *Description* field.

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



5 Click one of the following links to select a method for profile matching:

Match by Specific Element: Matching occurs using selected elements.

Match by Expression: Selects elements based on a regular expression or LDAP-style syntax. The match is based on the element's full distinguished name (DName), explained later in this section.

Match by Script: Selects elements based on a script written using the FormulaScript language. The entire text in the window must evaluate to either True or False for a given object. Only administrators with programming experience should use this feature. It is reserved for situations that require statements that exceed the complexity of an element expression, or that require validation of a property other than an element's DName.

Match by Class: Selects elements based on element class name.

For more information, see Section 11.2, "Determining an Element's Class Name," on page 150.

NOTE: The methods used to specify profile matching criteria are the same methods used to add elements to a Service Model. For details on the *Match By* options, see the *Operations Center 5.5 Service Modeling Guide*.

For profiles that query alarm history, select a top-level element to ensure alarm data collection for all levels in the hierarchy. Selecting a top-level element allows historical alarm queries to run at the top level or at any child level in the hierarchy. Do not use .* to collect data for all children. This results in many duplicate alarms in the *Historical Alarms* view in the Operations Center console.

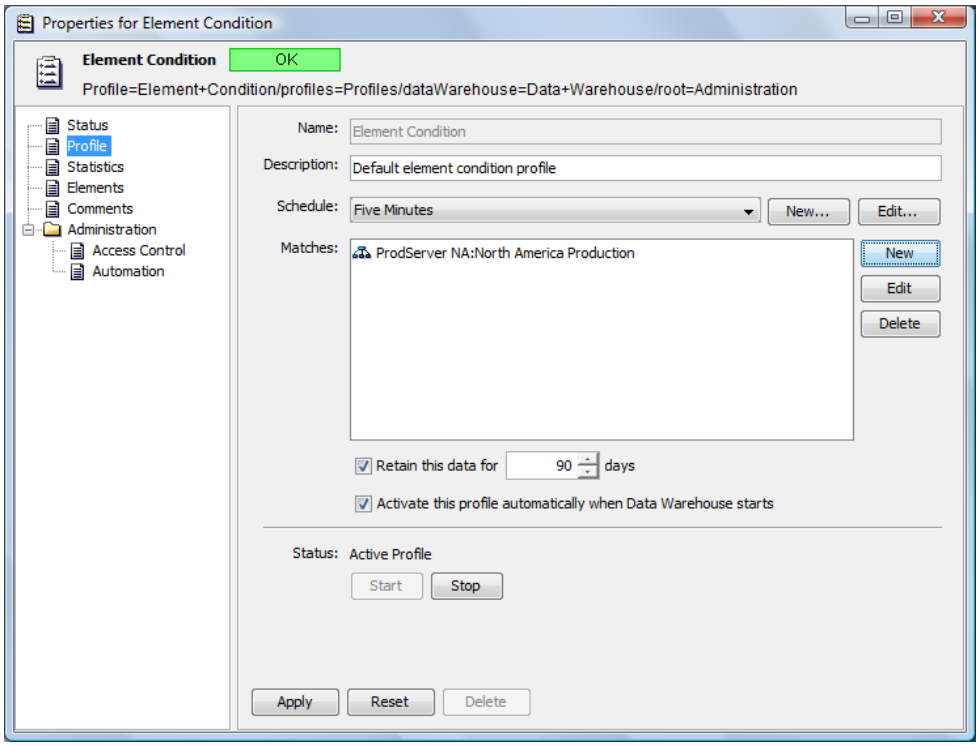
---

6 Click *Finish* to save the matches.

Editing Profiles

To update a profile:

1. In the *Explorer* pane, right-click a profile, then select *Properties* to open the *Status* property page.
2. In the left pane, click *Profile* to open its property page:



3. Edit the profile as needed.

   If data retention settings are changed, the new setting applies only to data collected from that point forward.
4. Click *Apply*.
5. To start the profile, click *Start*.
6. If the data retention setting was changed, stop and restart the Data Warehouse for changes to take effect.

   For instructions on how to stop and start the Data Warehouse Engine, see Section 12.5.3, "Stopping and Starting Data Collection," on page 194.

Deleting a Profile

When a profile is deleted, all its corresponding data is deleted during normal purge operations using data retention rules set at the time it was originally written to the database.

To delete a profile:

1 In the *Explorer* pane, right-click a profile element, then select *Delete Profile*.

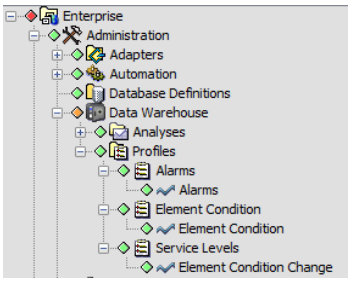
12.2.2 Creating Expressions

Expressions specify the types of data that are saved for future analysis. For example, an expression can be specified to save specific element attributes or children alarm information.

WARNING: If profiles are not deleted using the method listed above and the Operations Center software is uninstalled and the `OperationsCenter_install_path` directory is deleted, the data corresponding to the profiles remains in the database even though the profiles no longer exist. Also, expressions for invalid profiles can be listed in the *Property* page of the *Performance* view and Performance Analysis window.

One or more expressions must be defined for each profile to identify the type of data to save for elements that are selected using the profile's matching criteria. Operations Center provides a default profile (Alarms) that has a predefined Alarms expression.

Figure 12-2 *Explorer Pane: Expressions are listed beneath Profiles in the Explorer Pane.*



There are two basic types of expressions:

- "Alarm History Expressions" on page 185

  Alarm history expressions store historical alarms that display in the *Alarms* view.

- "Creating Historical Performance Expressions" on page 187

  Historical performance expressions capture performance data that display in the *Performance* view or Performance Analysis window.

WARNING: Do NOT set up more than one profile to measure the same alarm information (using the Alarms expression) for an element. Duplicate alarm data displays in the *Alarm Properties History* tab. However, more than one profile can be set up to measure alarm counts and all other performance indicators found in the *Historical Performance* section of the Create Expression dialog box.

Alarm History Expressions

An Alarms History expression creates a historical alarms data store that is accessed in the *Alarms* view by using the *Historical: Alarms* option.

Create an Alarm History expression adding one or more filters for the following items:

- Element name
- Severity
- Date/time
- Alarm property

  Specify a value and an operator such as:

  less than
  greater than
  contains
  does not contain
  is
  is not
  begins with
  ends with

  For example, in an Alarm History expression, create one filter to select alarms whose severity is CRITICAL and create another filter to select alarms whose property `TimeToRespond` is greater than 60 minutes.

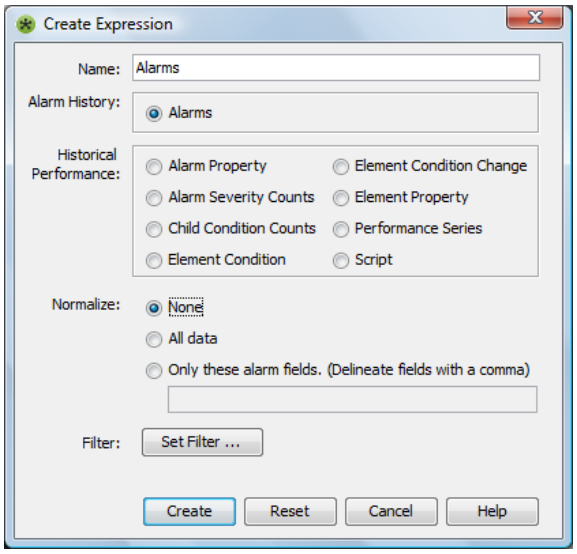
After adding a filter, select either *Match Any* or *Match All* to determine how the expression applies the filters:

- *Match Any* stores the data if any of the filter criteria are met
- *Match All* stores the data only if all filter criteria are met

To save alarm data in order to view alarm history in the *Alarms* view, create a profile that has an Alarms expression (Alarm History) attached to it.

To create an expression for gathering Alarm History:

1 In the *Explorer* pane, right-click a profile, then select *Create Expression* to open its dialog box:



2 Specify a name for the expression in the *Name* field.

Use a unique name for each expression of the same type.

3 To create an Alarm History expression, select the *Alarms* radio button.

Alarm history-related options display in the bottom of the Create Expression dialog box.

4 To set normalization rules, select one of the following *Normalize* radio buttons:

None: No alarm history data is normalized.

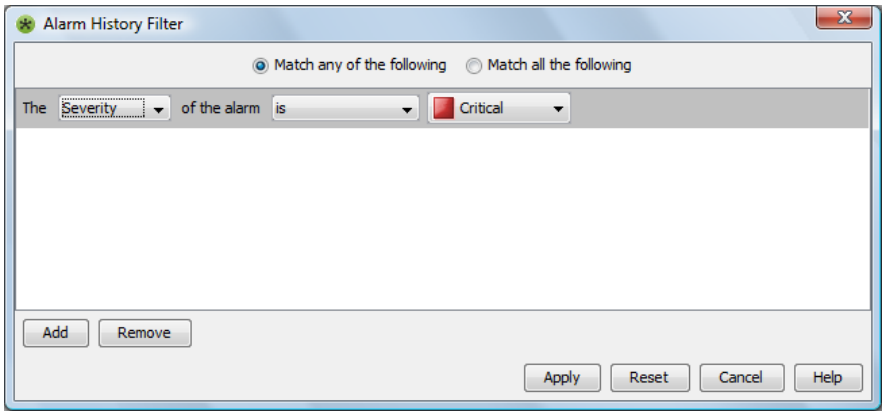
All Data: All alarm history data is normalized. Enabling this field causes a high volume of database activity that can significantly impact server performance. This rule is recommended only if alarm data is accessed using external reporting tools.

Only These Alarm Fields: Only specified alarm data is normalized when captured. Specify alarm column names in the field under the radio button. Separate entries with a comma.

NOTE: Normalization stores the alarm data in an expanded format in the database for easy data retrieval by external reporting tools.

---

5 To set up a filter for saving alarm data, click *Set Filter* to open the Alarm History Filter dialog box:



6 Specify the filter by using a combination of selections from the drop-down lists.

For example, create a filter that selects alarms whose severity is CRITICAL.

7 To specify an alarm property, select *Property* from the drop-down list, then specify the property name in the adjacent field.

8 Specify an operator, such as greater than, less than, or one of the following options, then specify a value.

9 Click *Add* to create another filter.

For example, create one filter to select alarms with CRITICAL severity and create another filter to select alarms whose property `TimeToRespond` is greater than 60 minutes.

To create the filter, do the following:

9a To remove a filter entry, select row, then click *Remove*.

The row is removed.

9b To save filter settings, click *Apply* to close the Alarm History Filter dialog box.

10 Select either the *Match Any* or *Match All* radio button to specify how to apply multiple filters.

Match Any stores the data if any of the filter criteria are met, while *Match All* stores the data only if all filter criteria are met.

11 Click *Create* to add the expression to the profile.

The expression displays as a child element of the profile. It can be edited later in the *Portal* view or the expression Properties dialog box.

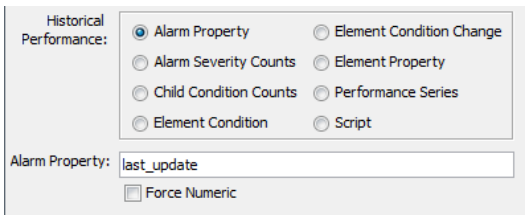
Creating Historical Performance Expressions

Historical Performance expressions are used to capture data that provide useful performance information. Historical Performance expressions are available as chart properties for associated elements. They display in the *Properties* pane found in the *Performance* view and Performance Analysis window.

Historical Performance expressions have data types that determine which data is saved as historical performance data and used to analyze single elements for performance charts. Performance charts (viewed in the *Performance* view or the Performance Analysis window) can only include the data specified by the Historical Performance expression data types. Additional data stored in database tables can be accessed directly through third party reporting tools, provided that a valid database driver (such as ODBC or JDBC) is available from the database vendor. For more information on the database tables, see the *Operations Center 5.5 Service Warehouse Data Dictionary*.

To create a Historical Performance expression:

1 Select a data type listed for Historical Performance in the Create Expression dialog box:



2 Configure a data type for a Historical Performance expression:

Alarm Property: A property from the *Alarms* view, such as date/time, priority, class. The list of valid property names varies among adapters. Use the same spelling (including underscores) that is used in the alarm column headings displayed in the *Alarms* view or in the alarm property pages. Some commonly used alarm columns:

- *Severity* to obtain the alarm's severity
- *ID* to obtain the alarm's ID
- *Last_Update* to obtain the most recent update time for an alarm
- *Persistent_ID* to obtain the persistent identifier for the alarm

To specify that the incoming string values are treated as numerals, select the *Force Numeric* check box.

Alarm Severity Counts: The total number of alarms by severity for a selected elements.

Child Condition Counts: The condition codes (critical, major, minor, and so on) of all child elements for a selected element.

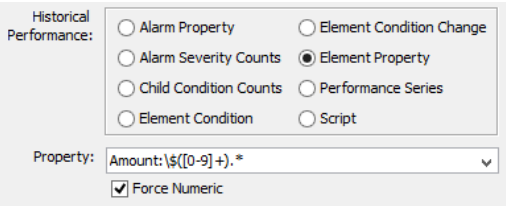
Regardless of Data Warehouse settings, the suppressed state is stored for all Historical Performance profiles using this expression, as well as the Element Condition, and Element Change Condition expressions. The exception is the Service Levels profile, which stores real time state only.

Element Condition: The condition code (critical, major, minor, and so on) for a selected element.

Element Condition Change: Any change in condition for a selected elements. Select the *Store Root Cause* check box to capture information regarding the root cause for the condition change. If capturing root cause, specify a severity level to act as a trigger from the *Condition Threshold* drop-down list.

Element Property: The values of a selected element property. Click the *Property* drop-down list, then select the name of an element property to monitor, such as memory usage or response time. Note, this list is only populated when the profile's elements have custom properties, meaning properties other than Element Condition, Last Reported, and Element Name.

A regular expression can be used to extract and retain a specific portion of the property value based on a pattern. This is most helpful when additional characters are present and you can only chart the numerical portion of the value. For example, values such as `43%`, `$35USD`, and `Rate is 88`; can be pruned to capture only the numbers.

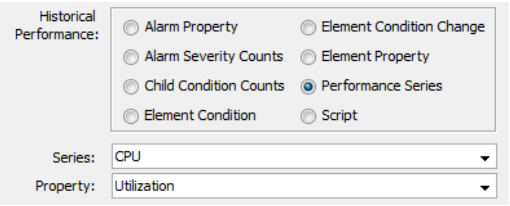


Append the property name with the regular expression using the format *`property_name:regex_pattern`* to prune property values. You'll notice in the following examples that the values to keep (any numbers from 0-9) are indicated inside of parentheses:

`Count:#([0-9]+)`	Ignores the `#` sign prior to the number in the *Count* property. For example, extracts 23 from #23, and 1 from #1
`Rate:Rate is ([0-9]+)`	Ignores the `Rate is` text before the number in the *Rate* property. For example, extracts `71` from `Rate is 71`, and 88 from `Rate is 88`.
`Amount:\$([0-9]+).*`	Ignores the `$` sign prior to the number and anything after the number in the *Amount* property. For example, extracts 25 from `$25usd`, and `400` from `$400 msrp`.

If numeric text is extracted, select *Force Numeric*.

Performance Series: Data derived from measurements made by an external management system:



Select a name for the series from the *Series* drop-down list and then specify the property to monitor in the *Property* drop-down list. The values vary among different management systems. By default, the expression Name value is created using the Series value and the Property value, separated by a period (i.e. Series.Property). The default Name value is editable.

Wildcards can be used. For example, enter the asterisk (*) as the Series and Property values. If an object has multiple series, they are all stored.

Script: The values resulting from running a script. Enter the entire contents of a script (written using FormulaScript) that determines the type of data to include.

12.2.3 Starting and Stopping Profiles

If you do not select the option to activate the profile when the Service Warehouse starts, then you can start it manually. Expressions must be added to a profile before it can be started. It is necessary to define expressions before starting a profile.

TIP: Stop and restart the related profile immediately after modifying a schedule, calendar, or expression.

The following sections explain the various ways to start and stop profiles:

- "Stopping or Starting All Profiles" on page 190
- "Stopping or Starting a Specific Profile" on page 190
- "Stopping the Profile and Ejecting the Queued Writes from the Repository" on page 190
- "Understanding When the Alarm Profile Stops Running" on page 190

Stopping or Starting All Profiles

1. In the *Explorer* pane, *Administration* > *Data Warehouse*.
2. Right-click *Profiles*, then select one of the following options:
   - *Start All Profiles*
   - *Stop All Profiles*

Stopping or Starting a Specific Profile

NOTE: Do not use the *Stop Profile* option if there are many queued "writes" to the repository of time series information for a profile because of a configuration error or over-matching of elements. Instead, use the *Stop and Purge Queue* option to stop the profile and eject the queued writes.

To start or stop a profile:

In the *Explorer* pane, right-click a profile, then select *Start Profile* or *Stop Profile*

The selected profile starts or stops respectively.

Stopping the Profile and Ejecting the Queued Writes from the Repository

Right-click the profile, then select *Stop and Purge Queue*.

The profile stops and the queue empties.

Understanding When the Alarm Profile Stops Running

When the Alarms profile is not running, the following message displays in the Operations Center trace file:

```
WARN Performance.Engine - Not storing annotation alarm; default alarm profile is
not activated.
```

If the default Alarm profile stops, alarm comments are not stored.

The Service Warehouse uses a file repository to keep temporary backups of data when the queue becomes too large. If a configuration error or over-matching of elements occurs, you can stop and purge the queue. This does not remove any data that is written to the backup repository files at the time the option is issued.

12.2.4 Monitoring Profile Data Collection

While a Data Warehouse Profile is collecting data, open the profile's Properties dialog box to view data collection statistics.

To view profile statistics and matched elements:

1. In the *Explorer* pane, right-click a profile, then select *Properties* to open the *Status* property page.
2. In the left pane, click *Statistics* to open its property page.

   Based on the interval selected for the profile, the following data updates while the profile is collects data:

Statistics	Description
Data Stored	The estimated size of the total data stored.
Data Average	The running average per minute of data stored.
Transactions	The number of write transactions to the database.
Pulses	The number of times the profile has performed data collection according to the schedule settings.
Errors	The number of errors that occurred upon storage.
Matched	The number of elements that match the profile.
Matching ms	The amount of time, in milliseconds, spent on executing pattern matching to find associated elements.
Measuring ms	The amount of time, in milliseconds, spent on measuring data values for the matched elements.
Storing ms	The amount of time, in milliseconds, spent on database call time to store the measured data values.

3. Click the *Elements* tab.

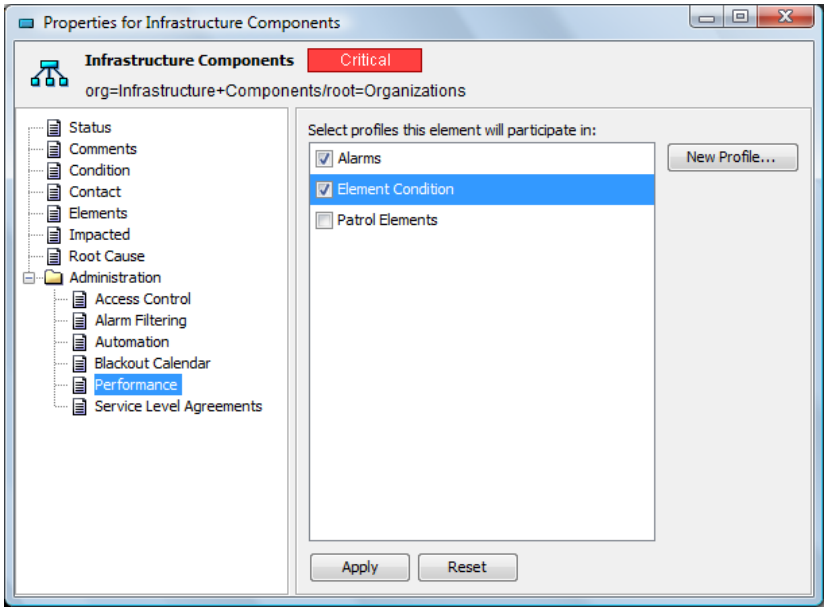
   The *Elements* tab lists all elements that currently match the profile.

12.3 Assigning Profiles at the Element Level

After profiles are started and elements are matched for performance and/or alarm history monitoring, it is possible to view the profiles in which a particular element participates. It is also possible to assign a profile to an element through the element's *Performance* property page.

To view the profiles in which an element participates:

1. From the *Explorer* pane, right-click an element, then select *Properties* to open the *Status* property page.
2. In the left pane, click *Performance* to open its property page:



   Profiles with a marked check box collect data for the element.
3. To add the element to a profile, select the check box next to the profile.
4. To detach a profile from an element, deselect the check box next to the profile name.
5. Click *Apply* to save the changes.

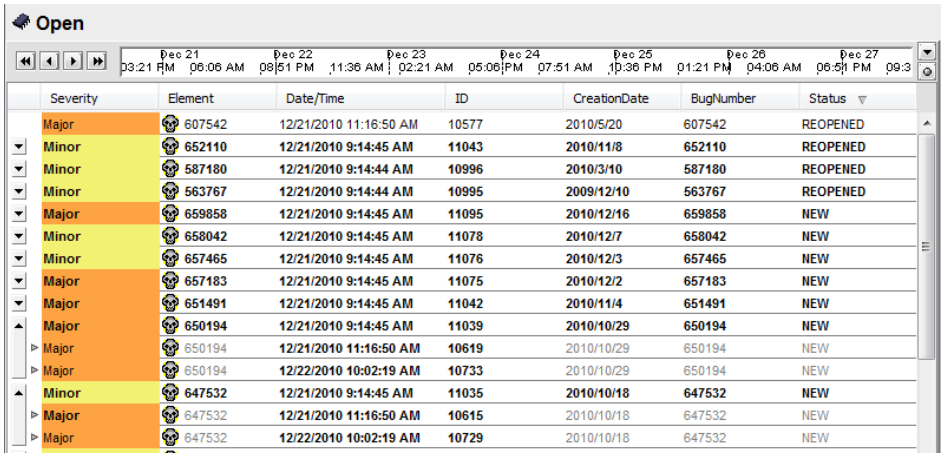
12.4 Viewing Alarm History

After capturing and storing alarm history, do one of the following to view the alarm history:

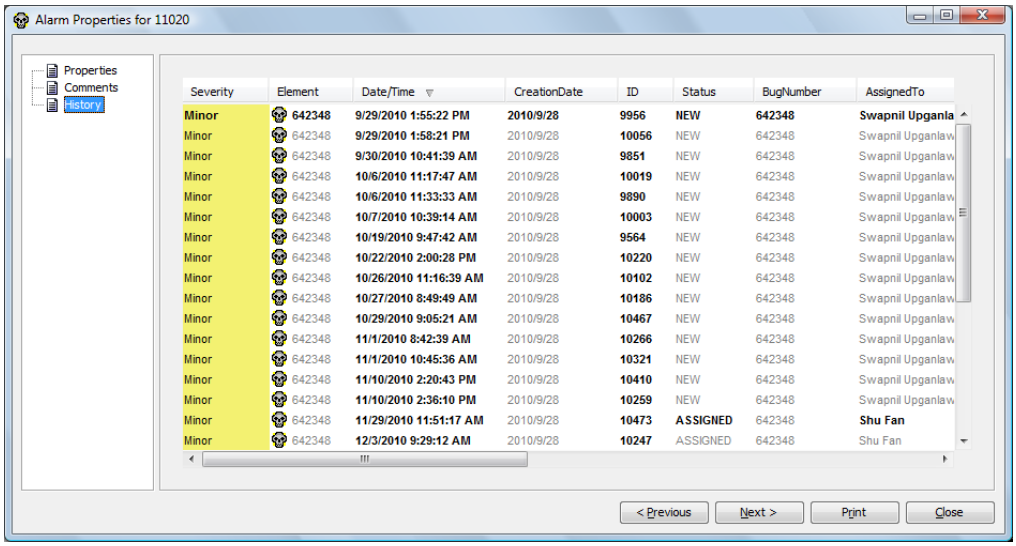
- View alarm history for elements by using the *Historical: Alarms* option in the *Alarms* view, then change the alarm display using the options listed under the *Format* menu.

  These options include defining the maximum number of historical alarms displayed.
- Double-click an alarm recorded, then select the *History* tab to view the history of a single alarm.

- Click the down arrow to view additional instances of an alarm:



- View the properties and history of an alarm using the alarm property pages:



12.5 Monitoring Alarm History Collection

Some useful information to consider after the alarm history data collection system is up and running includes implementing query time limits and how to stop and restart data collection:

- Section 12.5.1, “Changing Query Limit Values,” on page 194
- Section 12.5.2, “Changing Database Query Time Limits,” on page 194
- Section 12.5.3, “Stopping and Starting Data Collection,” on page 194
- Section 12.5.4, “Viewing Data Warehouse State and Statistics,” on page 194

12.5.1 Changing Query Limit Values

It is possible to change the alarm history query limit values by modifying the `/OperationsCenter_install_path/html/applet_params.xml` file.

To change the alarm history query limit values:

1. Open the `/OperationsCenter_install_path/html/applet_params.xml` file.
2. In the `<common>` section of the file, modify any of the following parameters:

   ```
   <param name="Alarm.HistoryLimitDefault" value="1000" />
   <param name="Alarm.HistoryLimitMax" value="50000" />
   <param name="Alarm.HistoryLimitPresets" value="500,1000,2500,5000,10000" />
   ```

   In this example, the default limit is 1000, the maximum limit is 50000, and the values displayed in the submenu are 500, 1000, 2500, 5000, and 10000.

12.5.2 Changing Database Query Time Limits

Operations Center allows database queries to run for up to 30 seconds before timing out.

To increase the query time limit, set a new `SLA.WatchdogLimit` value in the `Formula.custom.properties` file.

This property is set in milliseconds. The default value is 30000.

12.5.3 Stopping and Starting Data Collection

It is possible to manually stop and start the Data Warehouse Engine, which collects data for storage in the Service Warehouse. When the Data Warehouse Engine stops, all data collection for objects selected for data storage stops.

For example, if a system shuts down because of a problem, the Data Warehouse Engine stops collecting data for the objects selected for data storage. When the system is back up, it is necessary to use the *Start Data Warehouse* option to reactivate the Data Warehouse Engine and the SLM Engine.

To start or stop the Data Warehouse Engine:

1. In *Explorer* pane, right-click *Data Warehouse*, then select *Stop Data Warehouse* or *Start Data Warehouse*.

   The property pages for the *Database Warehouse* element display the status of the database, database connections, and cache statistics.

12.5.4 Viewing Data Warehouse State and Statistics

To view status and statistics for the Data Warehouse:

1. In the *Explorer* pane, right-click the *Data Warehouse* root element, then select *Properties*.

   The *Status* property page opens and displays information about *Repository Status*, *Queue Size*, and *Repository Details*.

   To disable text wrapping for easier reading, right-click the *Repository Details* pane, then select *Word Wrap*.
2. In the left pane, click *Data Warehouse* to open its property page.

3 To view warehouse settings and service level settings, click the *Data Warehouse*, *Backup Repository*, *System Settings*, or *Service Level Settings* tabs.

4 To view caching statistics for primary keys, alarm metadata, profiles, and time series information, click the *Cache Statistics* page.

To disable text wrapping, right-click the *Cache Statistics* pane, then select *Word Wrap*.

The cache statistics include the following information:

- Adapter Keys
- Alarm CRC Hash
- Alarm Meta Data
- Alarm Value Types
- Element Keys
- Profiles
- Root Cause Trees
- Root Cause Reasons
- Element Icons
- Time Band Keys
- Time Series Information
- Warehouse Series Information

12.6 Customizing Default Performance Chart Settings

Various properties can be set in the `applet_params.xml` to customize default chart settings such as suppressing end data point display, filling in time line holes as needed and setting the maximum outstanding requests to retrieve performance data.

To set custom properties for native performance data:

1 In a text editor, open the */OperationsCenter_install_path*/`html/applet_params.xml` file.

2 Do any of the following:

- To extend the last data point until the end of the time line window in numeric charts (not condition charts), add the following line:

  ```
  <param name="BSAChartPanel.extendLastDataPoint" value="true" />
  ```

- To control whether native performance data extends to the end of the chart time line by replicating the last data point AND fill time line holes as needed, add the following line :

  ```
  <param name="BSAChartPanel.nativePerformanceExtendAndFillHoles"
  value="true" />
  ```

  This setting overrides any value as set for the `BSAChartPanel.extendLastDataPoint` property above.

- To override the maximum number of requests queued per chart to retrieve performance data as the chart time line is adjusted and drawn, add the following line and specify the desired maximum number of requests to queue:

```
<param name="BSAChartPanel.requestQueueMaxSize" value="Max" />
```

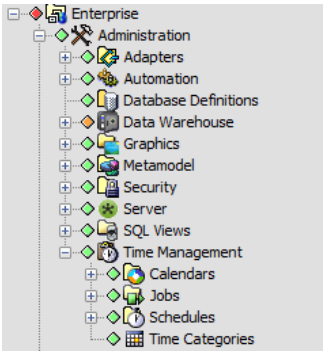
The system default is 4 outstanding requests. We recommend keeping the value low. Specifying zero or lower makes the performance data request queue unbounded.

3 Save the file.

13 Time Categories, Calendars, and Schedules

A major part of gathering business and performance data for service-level management is specifying when information is captured and processed. The Time Management objects that are used for configuring this are found under the *Administration* root element in the Operations Center console:

Figure 13-1 *Time Management Objects*



After setting up the appropriate date and time intervals, attach a schedule to a profile to define the dates and times for collecting alarms history and performance data. For more information on profiles and assigning schedules, see "Creating a Profile" on page 180.

Because of their interdependent relationships, you should create the Time Management objects in the following order:

1. Section 13.1, "Maintaining Time Categories," on page 198

   This section explains how to define specific labels for blocks of time. For example, a certain block of time can be designated as `maintenance` or `peak`.

2. Section 13.2, "Creating and Maintaining Calendars," on page 200

   This section explains how to define blocks of days and times when you want object monitoring to occur.

3. Section 13.3, "Creating and Editing Schedules," on page 209

   This section explains how to define larger time intervals for capturing data based on calendars.

For information about defining jobs and associating scripts to run during jobs, see the *Operations Center 5.5 Server Configuration Guide*.

13.1 Maintaining Time Categories

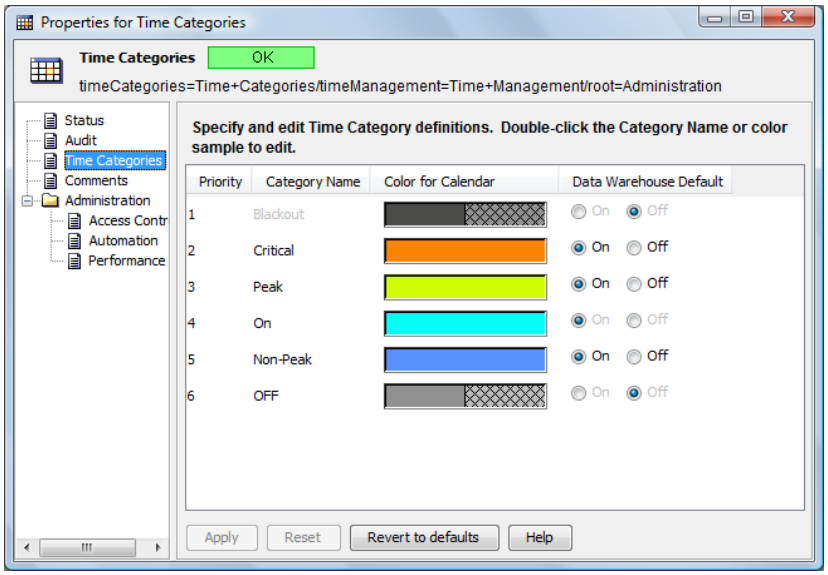
Time categories are used to identify important blocks of time. For example, peak hours can be defined as weekdays from 7 AM to 5 PM. Operations Center provides six default time categories, which can be customized:

Blackout
Critical
Peak
On
Non-Peak
Off

You can rank them by priority and select whether to enable or disable using them. By default, Blackout and Off are disabled.

To customize a time category:

1. In the *Explorer* pane, expand *Administration* > *Time Management*.
2. Right-click the *Time Categories* element, then select *Properties* to open the *Status* property page.
3. In the left pane, click *Time Categories* to open its property page:



4. To change the name of a time category, double-click the name in the *Category Name* column, specify a new name, then click anywhere in the dialog box to save the new name.

   The *Blackout* time category name cannot be edited.
5. To change the color, double-click the associated color box in the *Color for Calendar* column to open the Select a New Color dialog box, then do one of the following:
   - Use the *Swatches*, *HSB*, or *RGB* tabs to select the new color, then click *OK* to save the new color.
   - To close the dialog box without saving the color selection, click *Cancel*.
   - To revert to previous color settings, click *Reset*.

   The Time Categories Color Selector is the same as the one used to set up the condition and severity values.

6 (Optional) Consider whether to collect alarm history and performance data for each time category, then do one of the following in the *Data Warehouse* column:

- Select the *On* radio button to record alarm history and performance data in the Data Warehouse.
- Select the *Off* radio button to disable data capture of alarm history and performance data in the Data Warehouse.

7 To complete the customization process, do one of the following:

- To save new *Time Category* selections, click *Apply* to display a confirmation dialog box, then click *Yes* to save the changes.
- To revert to the default settings, click *Set to Defaults*.

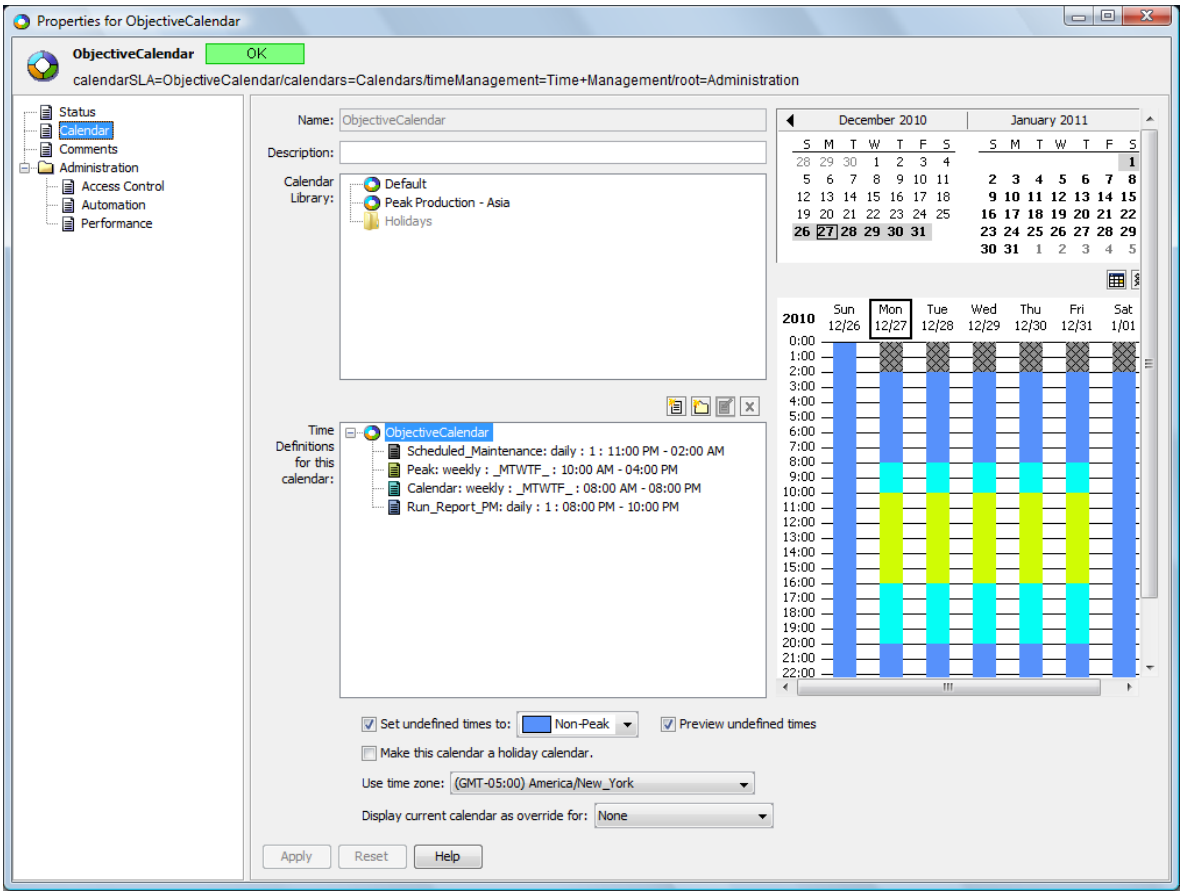
13.2 Creating and Maintaining Calendars

A calendar defines specific times and days for monitoring and capturing alarm history. Define a general calendar that captures data using the same days and times every month of the year, or define a more specific calendar that restricts data capture to specific months or days of the year.

Calendars consist of one or more time definitions. Each time definition specifies the days and time intervals for capturing data. You can define new time intervals for a calendar or copy existing ones from other calendars.

The Create Calendar dialog box contains a two-month calendar navigator (in the top right corner of the dialog box) for quick browsing and viewing of the scheduled data capture times for a specific date, as shown by colored blocks in the *Visualization* section directly below the calendar navigator:

Figure 13-2 *Calendar Properties*



The following sections explain how to create, configure, and maintain calendars:

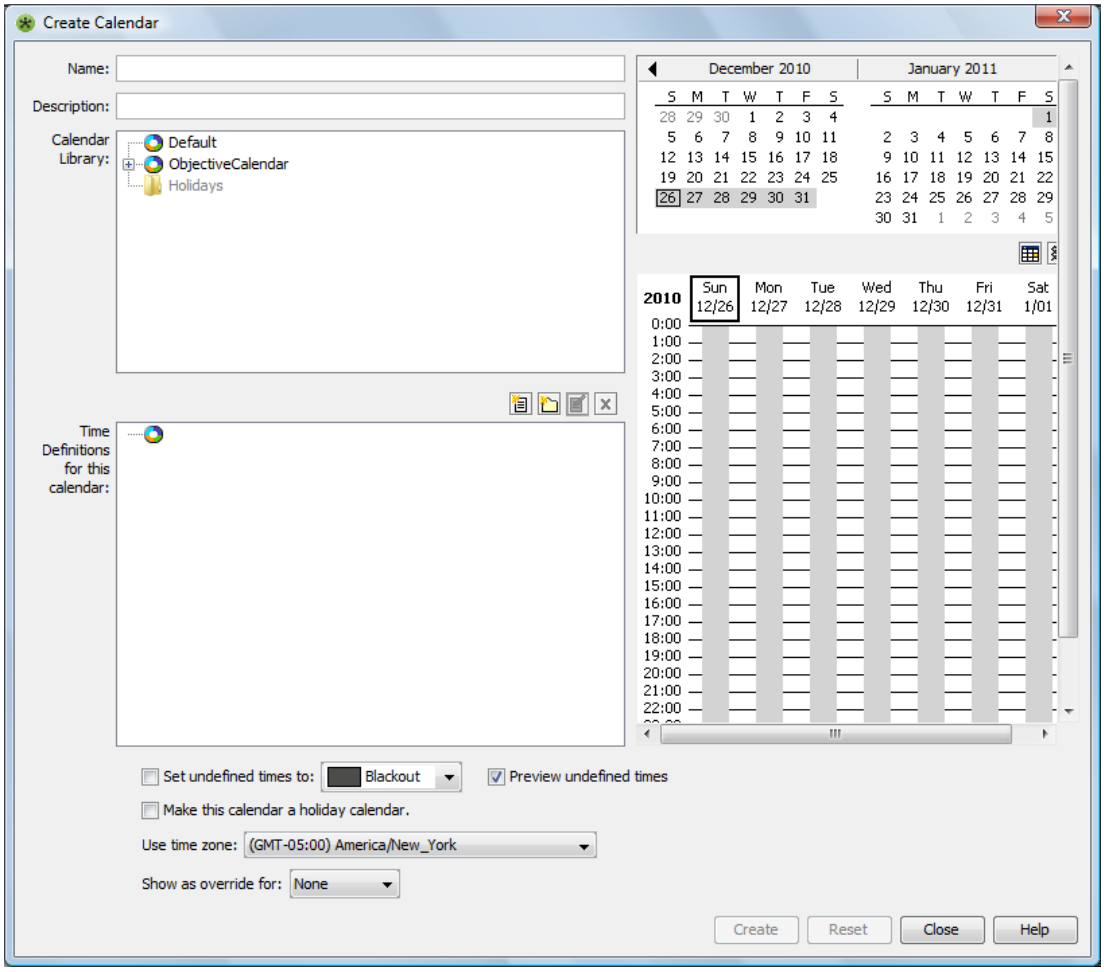
- Section 13.2.1, “Creating Calendars,” on page 201
- Section 13.2.2, “Specifying Time Definitions,” on page 202
- Section 13.2.3, “Using Existing Calendars to Create New Calendars,” on page 203
- Section 13.2.4, “Navigating the Calendar Visualization Section,” on page 204
- Section 13.2.5, “Assigning Undesignated Times,” on page 205
- Section 13.2.6, “Editing Calendars,” on page 206
- Section 13.2.7, “Setting Blackout Calendars on Elements,” on page 208

13.2.1 Creating Calendars

You can create calendars to associate with profiles used to select elements for which performance and alarm data is collected. For more information on profiles, see Section 12.2, “Using Profiles and Expressions to Capture Alarm History,” on page 178.

To create a calendar:

1. In the *Explorer* pane, expand *Administration* > *Time Management*.
2. Right-click the *Calendars* element, then select *Create Calendar* to open the Create Calendar dialog box.



3. Enter text in the calendar *Name* and *Description* fields.
4. To define the dates and times in the calendar, use the instructions in the following sections:
   - Section 13.2.2, “Specifying Time Definitions,” on page 202
   - Section 13.2.3, “Using Existing Calendars to Create New Calendars,” on page 203

5 Make selections for the remaining calendar options:

Set undefined times: For instructions, see Section 13.2.5, “Assigning Undesignated Times,” on page 205.

Make this calendar a holiday calendar : Select the check box to add the new calendar to the *Holidays* folder in the Calendar Library.

Use Time Zone: Select the time zone for the start and end times for the calendar.

Show as Override for: Merges a view of the current calendar with another calendar selected from the drop-down list. The time definitions of the selected calendar display in a merged view with the current calendar. This is for viewing purposes only. The merged view cannot be saved.

6 Click *Create* to save the settings and create the calendar.

The new calendar is added under the *Calendars* element in the *Explorer* pane.

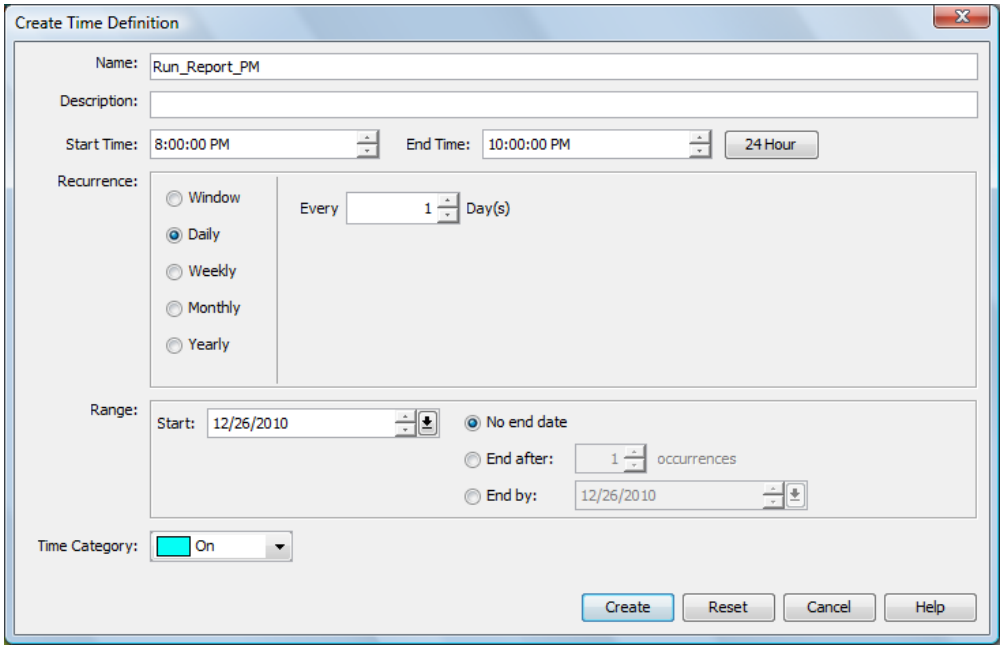
13.2.2 Specifying Time Definitions

In order to define a calendar, consider the days and times when alarm history and performance data should be captured. These time definitions can be created for a calendar or can be copied from an existing calendar.

Concerning defining overnight time definitions, if a time definition is set up to start on one day and finish sometime the next day (it runs overnight), on the first day the calendar is used, it starts at the start time specified by the time definition and there might be no overnight data collection. For example, assume the time definition specifies a peak time from 8 PM until 5 AM. On the first affected day, the calendar starts running at 8 PM, but does not continue to run from midnight until 5 AM on that first day. To capture this early morning time, you must define a separate time definition.

To define a new time definition to the calendar:

1 In the *Time Definitions* section of the Create Calendar dialog box, click *New Time Definition* to open the Create Time Definition dialog box:



2 Enter the following information to define the time definition:

Name: Name for the time definition.

Description: Description for the time definition.

Start/End Time: Starting and ending times for the time definition.

Use 24 hour period: Select to set the time definition to run a full 24 hours from the start time. Start Time and End Time are reset to 12:00 AM.

Recurrence: Select the interval for applying the time definition; for example, every day, every 2 weeks, and so on. Select *Window* to select a specific start date/time and end date/time.

Range: Set the start date and optional ending date.

Time Category: Click the *Time Category* drop-down list, then select the appropriate category for the time definition (such as Peak, On/Off time, and so on).

3 Click *Create* to save the *Time Definition* as part of the current calendar.

It displays in the *Time Definitions for this Calendar* section of the dialog box.

13.2.3 Using Existing Calendars to Create New Calendars

Time definitions from existing calendars can be copied or linked to a new calendar.

Linked calendars are only editable in the original calendar definition and all changes to the original calendar affect all linked calendars.

To copy or link to an existing calendar:

1 In the *Calendar Library* section of the Create Calendar dialog box, navigate to a calendar to use as the basis for the new calendar.

2 Right-click the calendar or portion of a calendar (entire calendar, or a specific folder or time definition), then select *Copy*.

3 In the *Time Definitions for This Calendar* section, right-click and do one of the following:

- Select *Paste* to create an editable duplicate in the current calendar.
- Select *Paste Link* to create a link to the original calendar.

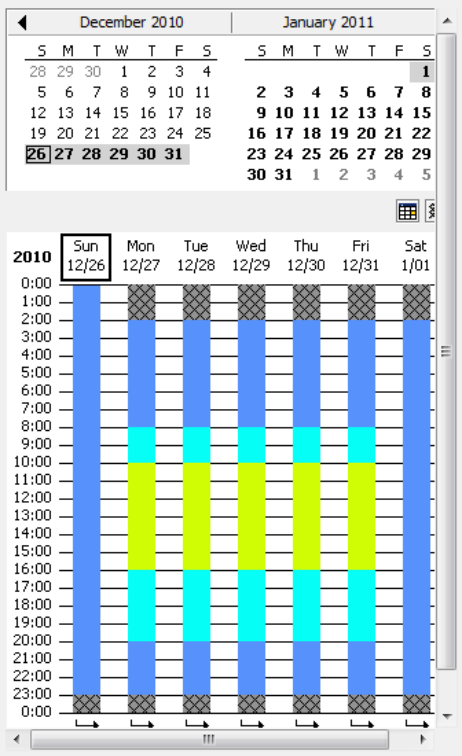
  A linked read-only item is created. Only the original can be edited.

13.2.4 Navigating the Calendar Visualization Section

The *Calendar Visualization* section on the right side of the *Calendar* page consists of a two-month calendar and a weekly calendar. The weekly calendar displays shaded blocks of time to represent time definitions for the selected week. The colors correspond to the time definition category (Peak, On, Critical, and so on).

Figure 13-3 illustrates how you can view the time definitions as colored blocks in the weekly calendar:

Figure 13-3 *The Calendar Visualization Section of the Create Calendar Dialog Box*



Tasks that you can perform in this dialog box:

- "Navigating the Two-Month Calendar" on page 204
- "Viewing Shaded Blocks That Represent All Time Definitions in the Calendar" on page 205
- "Viewing Only One Time Definition" on page 205
- "Returning to the Current Date in the Visualization Window" on page 205
- "Displaying or Hiding Times When SLM Data Collection Is Disabled" on page 205

Navigating the Two-Month Calendar

1. Click the right and left arrows (◀ and ▶) to display previous or next month.
2. Click a day in the calendar to display the associated week in the detailed calendar section below the two-month calendar.

Viewing Shaded Blocks That Represent All Time Definitions in the Calendar

Select the calendar name in the *Time Definitions for This Calendar* section.

Viewing Only One Time Definition

Select a time definition name in the *Time Definitions for This Calendar* section.

Returning to the Current Date in the Visualization Window

Click *Go to Today* to display the current date.

Displaying or Hiding Times When SLM Data Collection Is Disabled

Click *Show/Hide BSA Off Times*.

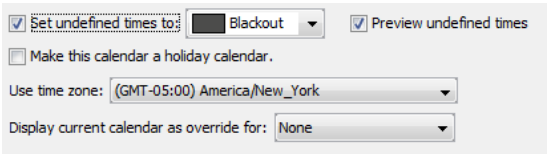
When enabled, a black grid displays in the calendar to identify when SLM data collection is turned off.

13.2.5 Assigning Undesignated Times

After designating some specific time periods, it is possible to assign all remaining undefined times to a specific time category. For example, if the time period 12 AM to 5 AM is not associated with a time definition, it can be assigned to an existing time category, such as `Blackout`.

To assign all undesignated time in a calendar to a specific time category:

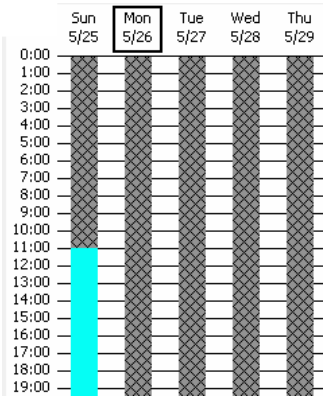
1. Select the *Set Undefined Times* check box.
2. Click the drop-down list, then select a time category:



3 Select the *Preview Undefined Times* check box.

The calendar columns update to show the undefined time color.

For example, in the following figure, the undefined times are assigned to the `Blackout` time category and are shaded black in the calendar:



13.2.6 Editing Calendars

The Default calendar cannot be edited or deleted. You can do the following to edit calendars:

- "Editing an Existing Calendar" on page 206
- "Editing a Time Definition or Folder" on page 206
- "Deleting a Time Definition or Folder" on page 207
- "Adding a New Folder to the Calendar" on page 207
- "Editing Calendar Time Intervals" on page 207

Editing an Existing Calendar

To edit a calendar:

1 In the *Explorer* pane, expand *Administration* > *Time Management* > *Calendars*.
2 Right-click a calendar, then select *Properties* to open the calendar's *Status* property page.
3 In the left pane, click *Calendar* to open its property page.
4 Edit the calendar as needed.
5 Click *Apply* to update the calendar.

Editing a Time Definition or Folder

To edit a time definition or folder:

1 In the *Time Definitions* section of the *Calendar* property page, select the time definition or folder, then click *Edit* to open the Edit Time Definition dialog box.
2 Edit the time definitions information as needed.
3 Click *Apply* to update the time definition.

Deleting a Time Definition or Folder

To delete a time definition or folder:

1. In the *Time Definitions* section of the *Calendar* property page, select a definition or folder, and then click *Delete* to open a confirmation dialog box.
2. Click *Yes* to confirm the deletion to remove the definition or folder.

Adding a New Folder to the Calendar

To add a new folder:

1. In the *Time Definitions* section, click *New Folder* to open the Create Folder dialog box.
2. Enter a folder name, then click *OK*.

   The new folder displays in the *Time Definitions* section.

Editing Calendar Time Intervals

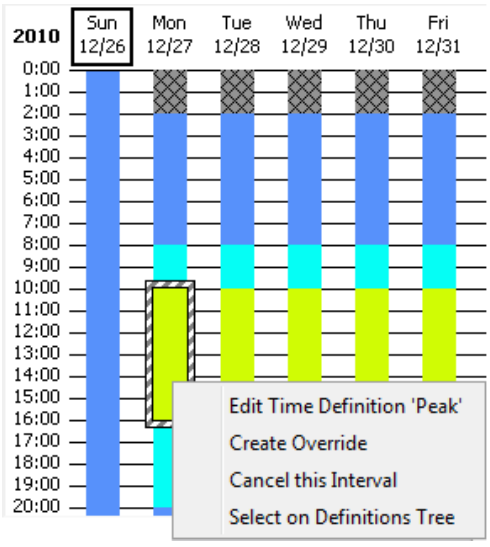
Individual time intervals displayed in the weekly calendar section of the Create Calendar or Edit Calendar dialog box can be modified.

You can do the following to edit calendar time intervals:

- "Modifying an Individual Time Interval" on page 207
- "Editing a Time Interval or Time Definition Directly" on page 208

Modifying an Individual Time Interval

In the calendar visualization section of the *Calendar* property page, right-click a time interval, then select an option:



Edit Time Definition: Enables editing the time definition associated with the selected time interval. Note that all changes made apply to all dates affected by the time definition.

Create Override: Enables changing the start and/or end time or time category for the time interval. To remove the override later and use the default interval defined by the time definition, right-click the interval, then select *Remove Override*.

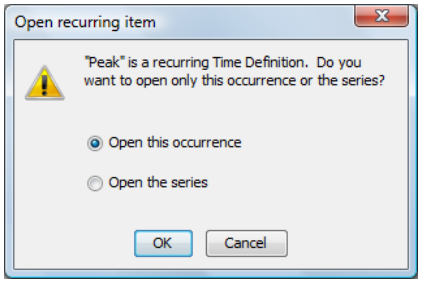
Cancel this Interval: Removes the time interval from the time definition for the selected date only. To restore the interval, right-click the interval, then select *Restore Cancelled Interval*.

Select on Definitions Tree: Highlights the time definition associated with the interval. This is useful when many time intervals and definitions exist.

Editing a Time Interval or Time Definition Directly

To edit a time interval or definition:

1. In the calendar visualization section of the *Calendar* property page, double-click a time interval to open the Open Recurring Item dialog box:



2. Do one of the following:
   - Select the *Open This Occurrence* radio button to edit the start and/or end time or time category associated with this interval only.
   - Select the *Open the Series* radio button to edit the time definition associated with the time interval.
3. Click *OK* to open the Edit Time Definitions dialog box.
4. Edit the Time Category and start and end times as needed.
5. Click *Apply* to update the time interval.

13.2.7 Setting Blackout Calendars on Elements

It is possible to apply a blackout calendar only at the element level for selected elements. Settings in the element's blackout calendar override all other calendar settings applied to that element.

However, as an exception to the rule, data is collected during all blackout calendars when capturing SLA data, to ensure that data is available in the event that blackout periods change after data capture. In this case, the blackout data can be excluded when running SLA reports. For more information, see the *Operations Center 5.5 Service Level Agreement Guide*.

To create a blackout calendar for a specific element:

1. In the *Explorer* pane, navigate to the element for which a blackout calendar is needed.
2. Right-click the element, then select *Edit Blackout Calendar* to open its dialog box.

3 Create Time Definitions as necessary to specify blackout times for the element.

   For more information regarding creating calendars and defining time definitions, see Section 13.2.1, “Creating Calendars,” on page 201.

4 Click *Apply* to save the blackout calendar settings.

13.3 Creating and Editing Schedules

A schedule defines the time intervals for capturing history and performance data, based on a selected calendar.

For example, assume a calendar captures data every Monday and Friday in every month, from 3:00 PM – 6:00 PM. A schedule associated with this calendar can capture data at fifteen-minute intervals between 3 PM and 6 PM, Monday through Friday.

In addition, a second schedule can use the same calendar, but capture data at five-minute intervals during the time period defined in the calendar.

Default schedules (One Minute, Five Minutes, Audit Schedule) cannot be edited or deleted.

To create and manage schedules:

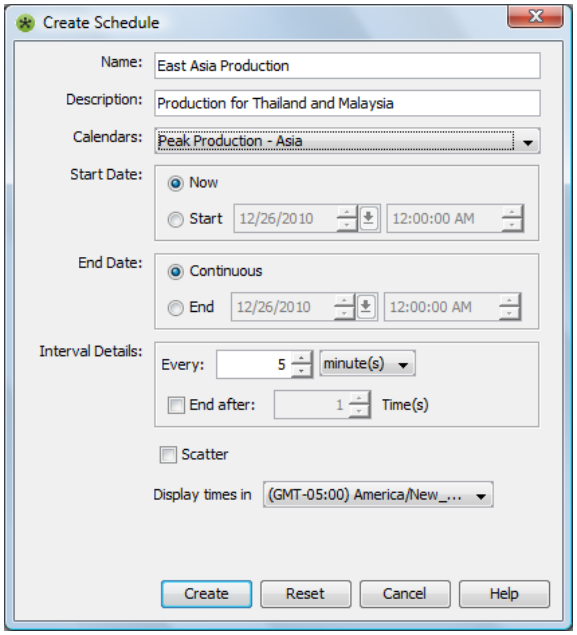
- Section 13.3.1, “Creating a Schedule,” on page 209
- Section 13.3.2, “Editing a Schedule,” on page 211
- Section 13.3.3, “Deleting a Schedule,” on page 212

13.3.1 Creating a Schedule

To create a schedule:

1 In the *Explorer* pane, expand *Administration* > *Server* > *Time Management*.

2 Right-click the *Schedules* element, then select *Create Schedule* to open its dialog box:



3 Fill in the fields:

Name and Description: Text naming and describing the schedule.

Calendar: Click the drop-down list, then select an existing calendar to apply to the new schedule.

Start Date: Set the dates when the schedule begins:

- Select the *Now* radio button to select the current date as the start date.
- Select the *Start* radio button and use the spinners to select the month, date, and time.
- Click the drop-down list to display a calendar. Select a date, then click the drop-down list again to close the calendar.

End Date: Set the dates when the schedule ends:

- Select the *Continuously* radio button to define no end date for the schedule.
- Select the *End* radio button and use the spinners to select the month, date, and time.
- Click drop-down list to display a calendar. Select a date, then click the drop-down list again to close the calendar.

Interval Details: Use the Every spinner to select a number, click the *Every* drop-down list, then select the type of interval. For example, every 35 minutes.

If the schedule should run only a specific number of times, select the *End After* check box and use the *End After* spinners to specify the number of times the scheduled process runs.

Scatter: Select the check box to allow the schedule to adjust randomly to maximize efficiency and minimize taxing system resources.

Display Times: Click the drop-down list, then select the time zone for the schedule. The time zone used for the schedule can differ from the one used for the associated calendar.

4 Click *Create* to create the schedule.

13.3.2 Editing a Schedule

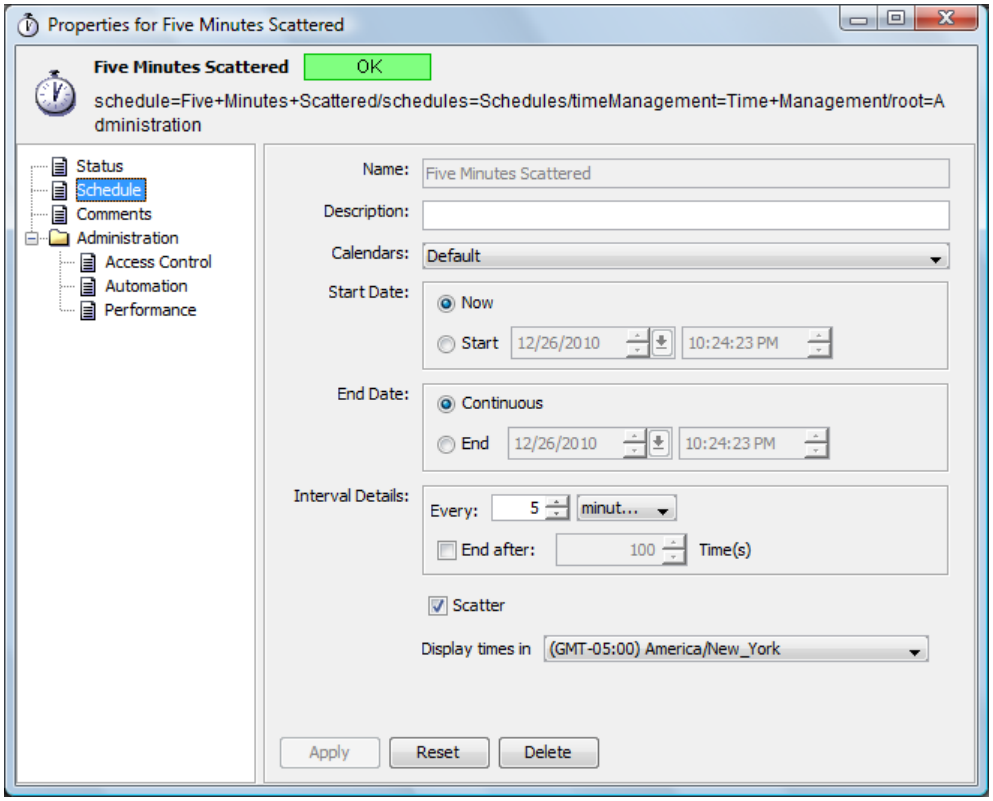
If a schedule is modified, stop and restart the associated profiles to update the profile. For details on profiles, see Section 12.2.1, "Creating Profiles," on page 178.

To edit a schedule:

1 In the *Explorer* pane, expand *Administration* > *Server* > *Time Management* > *Schedules*.

2 Right-click a schedule, then select *Properties*.

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



4 Edit the schedule as needed.

5 Click *Apply* to update the schedule.

13.3.3 Deleting a Schedule

Deleting a schedule removes it from all profiles to which it is attached, thereby affecting the dates and times that objects are monitored.

To delete a schedule:

1. In the *Explorer* pane, right-click a schedule element, then select *Delete Schedule* to open a confirmation dialog box.
2. Click *Yes* to confirm the deletion to delete the schedule.

14 Defining and Managing Automation Events

Automation events can trigger an alert when a network event occurs that might require intervention. Automation alerts can include audio signals, notifications sent via e-mail, or entering information in a database about an event.

Scripting capabilities are available to define more complex actions, such as tracking actions performed on alarms. For information about using scripts to access information about custom properties and classes, see the *Operations Center 5.5 Scripting Guide*.

An automation event is an action triggered by an activity or condition change that occurs in a network. Automation events are defined to notify the appropriate personnel that an event occurred and might require intervention. There are two types of automations:

- **Client-Side Automations:** Generally report information to a user or group.
- **Server-Side Automations:** Generally use scripts to integrate to back-end systems. For example, a new ticket might be opened in a trouble-ticketing system.

The user who creates an automation event is the event owner. The owner can configure different automation actions for the same conditions.

Automations are accessed via the *Automations* tab in the Properties dialog box for:

- **Element:** Automations run for the user that defines them.
- **User or Group:** Automations run on the selected element just for the users and/or groups they are setup for.
- **Automation Server:** Automations are sent to the server and can be directed to any destination.

Creating an automation event requires specifying and defining the following two components:

- **Automation Filters:** These define the conditions, states, system changes, or other information that changes about the system or system resources for which you want to launch a response or automation action.
- **Automation Actions:** These define the system response launched when a filter detects that an automation event occurred. Actions can include playing an audio file, firing a script, creating a log record, or sending an e-mail.

To define and manage automation events:

- Section 14.1, “Understanding Client-Side and Server Side Automations,” on page 214
- Section 14.2, “Defining Automation Events,” on page 216
- Section 14.3, “Monitoring and Managing Automation Events,” on page 218
- Section 14.4, “Defining Automation Events Directly on Elements,” on page 223
- Section 14.5, “Monitoring and Managing Automation Events for an Element,” on page 224

- Section 14.6, “Using Automation Filters,” on page 226
- Section 14.7, “Using Automation Actions,” on page 231

14.1 Understanding Client-Side and Server Side Automations

Automation events can be defined for a user, group, or the Automation server. Automation events associated with a group apply to the users of that group. Each automation event created for a user or group must specify a target element for the selected automation filter and action.

- **Client-Side Automations:** These automation events are configured specifically for a user or group.
- **Server-Side Automations:** The automation events are configured specifically for the Automation Server.

Both client-side and server-side automations are defined from the *Automation* element under *Administration*. For information on defining automations, see Section 14.2, “Defining Automation Events,” on page 216.

Automation events can also be defined at the element level, but are configured for the current user only. For instructions on defining automation events at the element level, see Section 14.4, “Defining Automation Events Directly on Elements,” on page 223.

Actions and filters are available depending on whether the automation is client-side or server-side. Table 14-1 shows which actions apply to client-side and server-side automations. Table 14-2 on page 215 shows which filters apply to client-side and server-side automations.

Table 14-1 *Actions available for Client-Side and Server-Side Automations*

	Available for Automations on...	
Action	**Client-Side**	**Server-Side**
Generate a computer noise	X	
Mail element and alarm information		X
Mail element information		X
Open alarm popup window	X	
Page element	X	X
Play an audio clip	X	
Post alarm to tec		X
Post to tec		X
Run a predefined script from the script library	X	X
Run a script	X	X
Start chirping	X	
Start the gong	X	
Stop a running audio clip	X	
Stop chirping	X	

	Available for Automations on...	
Action	**Client-Side**	**Server-Side**
Stop the gong	X	

Table 14-2 *Filters available for Client-Side and Server-Side Automations*

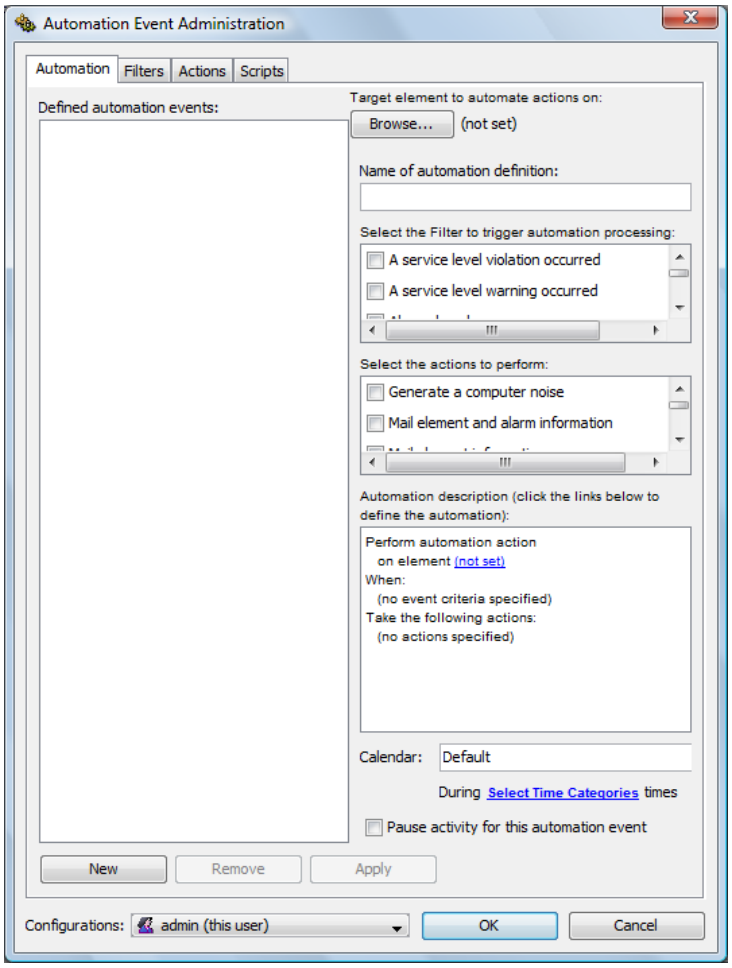
	Available for Automations on...	
Filter	**Client-Side**	**Server-Side**
A service level violation occurred	X	X
A service level warning occurred	X	X
Alarm closed	X	X
An alarm event occurs on the element	X	X
An alarm operation was performed		X
An element operation was performed		X
An element property was added/removed/changed		X
An element relationship was added/removed/changed		X
An element was created		X
An element was destroyed		X
Any alarm opened	X	X
Critical alarm opened	X	X
Element at OK condition	X	X
Element at critical condition	X	X
Element at information condition	X	X
Element at major condition	X	X
Element at minor condition	X	X
Major alarm opened	X	X
Minor alarm opened	X	X
The element's condition changes	X	X

14.2 Defining Automation Events

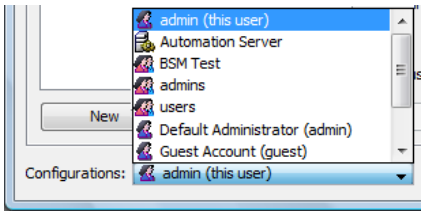
Configure automations for the Automations Server, and any user or group from the *Automation* element under *Administration*, rather than having individually configure automations from each user or group.

To define automation events:

1. In the *Explorer* pane, expand *Administration*.
2. Right-click *Automation*, then select *Properties* to open the *Status* property page.
3. In the left pane, under *Administration*, click *Automation* to open its property page.
4. Click *Change* to open the Automation Event Administration dialog box:

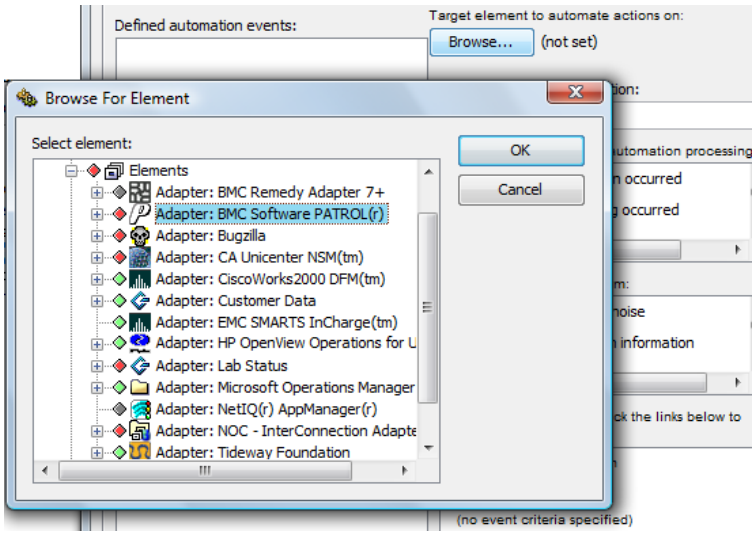


5. Click the *Configurations* drop-down list located at the bottom of the dialog box, then do one of the following:
   - To define server-side automations, select *Automation Server*.
   - To define client-side automations, select a user or group.

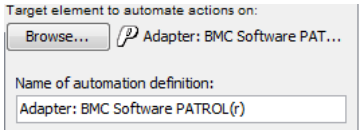


6 Click *New* to define a new automation.

7 Click *Browse* at the top of the dialog box to select a target element for the automation event.



The selected element displays in the *Name of Automation Definition* field and *Under Target Element to Automate*:



If you prefer a different name for the automation even, you can change the name in the *Name of Automation Definition* field.

8 In the *Select the Filter to Perform* section, select one or more check boxes to specify filters to trigger the automated action.

9 To define additional filters, click the *Filter* tab.

For more information about creating filters, see Section 14.6, "Using Automation Filters," on page 226.

10 In the *Select the Actions to Perform* section, select one or more check boxes to specify the actions to perform when the filter is applied.

11 To define additional actions, click the *Actions* tab.

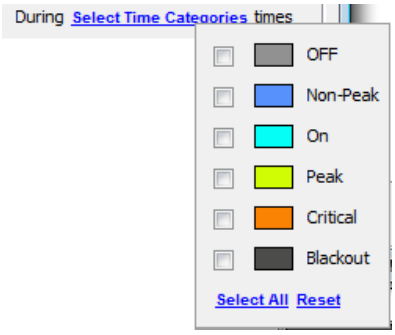
For information about creating actions, see Section 14.7, "Using Automation Actions," on page 231.

12 In the *Automation Description* section, click a link to view, then select additional parameters for the automation event.

13 Click the *Calendar* drop-down list, then select a calendar to identify when to watch for the filter events:

For information about calendars, see Chapter 13, "Time Categories, Calendars, and Schedules," on page 197.

14 Click the *During Times* link, then select one or more check boxes to specify applicable time categories within the calendar:



For more information about time categories, see Chapter 13, "Time Categories, Calendars, and Schedules," on page 197.

15 To pause the event, select the *Pause Activity for This Automation Event* check box.

16 Click *Apply*.

The automation event for the user, group, or Automation server is activated.

14.3 Monitoring and Managing Automation Events

After automation events have been defined and applied to a server, user, or group, you can monitor the status of each event, and manage the automation information:

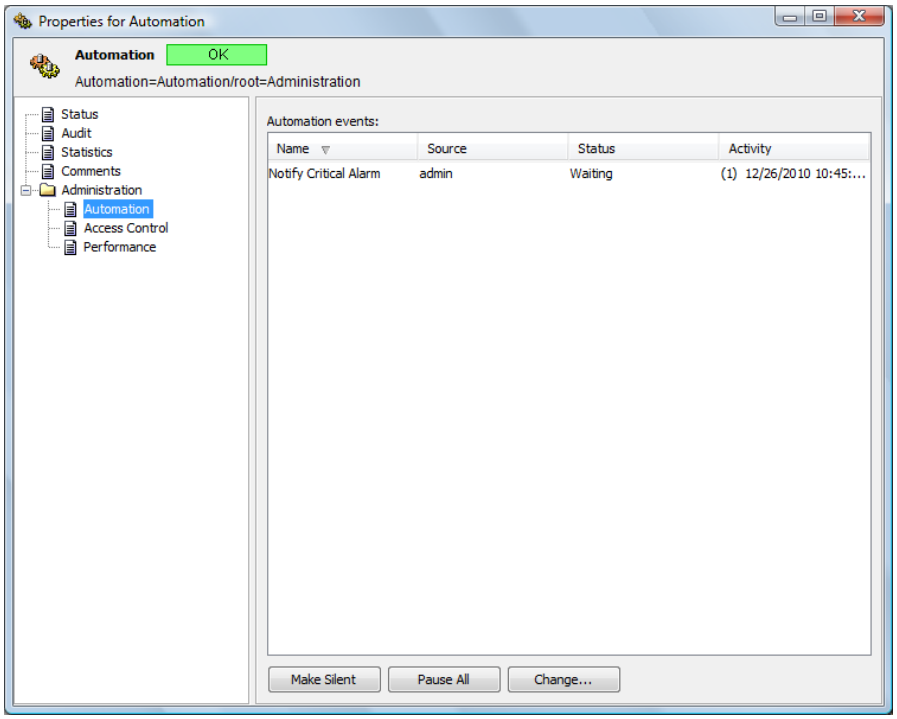
- Section 14.3.1, "Monitoring Your Client-Side Automation Events," on page 219
- Section 14.3.2, "Viewing Statistics about Server Side Automation Tasks," on page 220
- Section 14.3.3, "Viewing the Server-Side Automations Queue," on page 221
- Section 14.3.4, "Clearing the Server-Side Automations Queue," on page 221
- Section 14.3.5, "Canceling the Current Server-Side Automation Task," on page 222
- Section 14.3.6, "Editing Automation Events," on page 222

14.3.1 Monitoring Your Client-Side Automation Events

An Automation Events list allows you to view or monitor any client-side automation events created for your user account or any groups you are a member of.

To monitor automation events:

1. In the *Explorer* pane, expand *Administration*.
2. Right-click the *Automation* element, then select *Properties*.
3. In the left pane, click *Automation* to open the *Automation* property page:



The *Automation Events* section lists all of the automation events.

4. Review the information for the automation events in the four columns:

   Name: Name defined for the automation event.

   Source: User, group, or server for which the event is configured.

   Status: The status normally is `Waiting` for the next event to occur. If the automation has been paused, the status is `Paused`.

   Activity: Date and time and element DName of the most recent automation event.

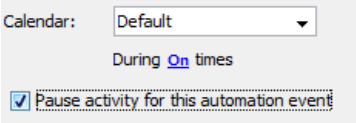
5. Do one of the following:

   - To silence the alarms for a selected automation, select the automation, then click *Make Silent*. The alarm is silenced and the button changes to `Make Audible`.
   - To halt all automation events, click *Pause All* to change it to `Unpause All`.
   - To pause a specific automation, select the automation, then click *Change* to open the Automation Event Administration dialog box.

6 To pause activity for the event, do the following:

6a Select the event and click *Change*. The Automation Event Administration dialog box displays.

6b Select the *Pause Activity for This Automation Event* check box.



6c Click *Apply*, then close the dialog box.

7 Click *Apply* to save the changes.

If activity is paused for an event, the event remains paused until you deselect the *Pause Activity for This Automation Event* check box, then click *Apply*.

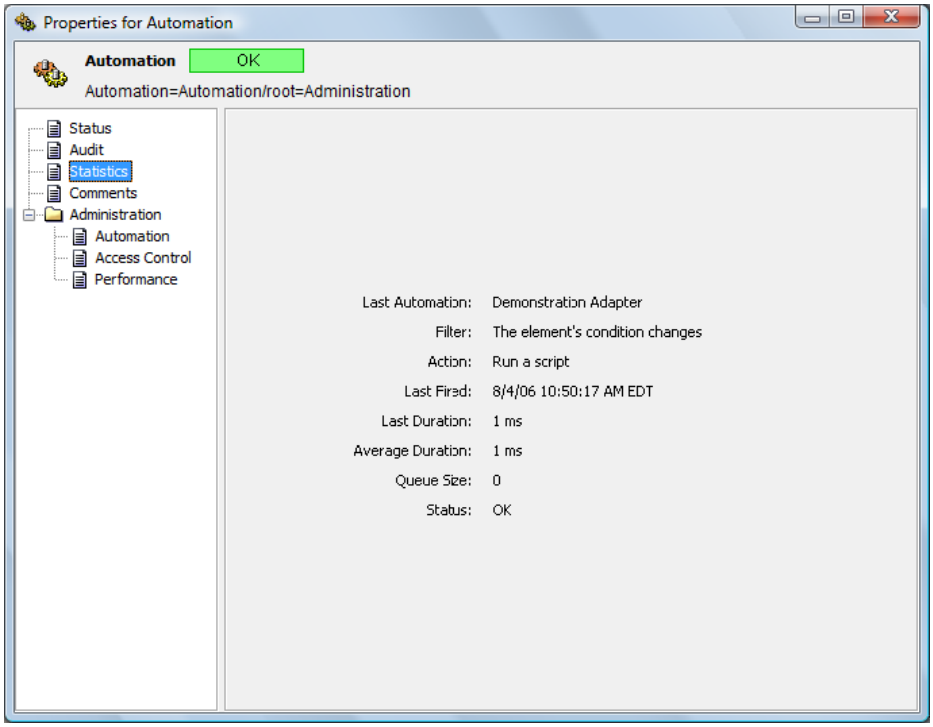
14.3.2 Viewing Statistics about Server Side Automation Tasks

The *Statistics* tab for the *Automation* element enables administrators to monitor server-side automations (script-based and nonscript-based), and if necessary, empty the queue or cancel problematic tasks.

To view statistics about server-side automation tasks:

1 In the *Explorer* pane, right-click the *Automation* root element, select *Properties* to open the *Status* property page.

2 In the left pane, click *Statistics* to open its property page:



3 Review the following statistics:

Last Automation: Name of the server-side automation that fired most recently.

Filter: The filter condition that caused the most recent automation to fire.

Action: The action performed by the most recent automation.

Last Fired: The date when the most recent server-side automation was fired.

Last Duration: The amount of time, in milliseconds, that the most recent server-side automation task was completed successfully.

Average Duration: The average amount of time, in milliseconds, that represents the duration of all server-side automation tasks since the Operations Center software was started.

Queue Size: The current number of entries in the server-side automation queue, excluding the automation that is currently executing (if any).

Status: The status of the automation that fired most recently. Values include:

- **OK:** The automation ran successfully.
- **Exception:** The automation ran but had exceptions.
- **Cancelled:** The automation was cancelled by the user.
- **Not Dispatched:** No automations dispatched at startup.

14.3.3 Viewing the Server-Side Automations Queue

Server-side automation tasks waiting to be executed are stored in a queue.

To view the automation tasks currently in the queue:

1 In the *Explorer* pane, right-click the *Administration* > *Automation* element, then select *View Automation Queue*.

The Automation Queue dialog box displays a list of automation tasks waiting to execute.

2 Click *Exit* to close the dialog box.

14.3.4 Clearing the Server-Side Automations Queue

Server-side automation tasks waiting to be executed are stored in a queue, which can be cleared.

To cancel all automation tasks in the queue:

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

2 Right-click *Automation* and then select *Flush Queue*.

All automation tasks waiting to execute are deleted.

14.3.5 Canceling the Current Server-Side Automation Task

The *Cancel Current* option can cancel the current server-side automation task.

To cancel an automation task:

1. Right-click the *Automation* element in the *Explorer* pane and select *Cancel Current* to stop the automation task

14.3.6 Editing Automation Events

To make changes to automation events:

1. In the *Explorer* pane, expand *Administration*.
2. Right-click *Automation*, then select *Properties* to open the *Status* property page.
3. In the left pane, click *Automation*.

   The *Automation* property page opens and contains a list of all automations.
4. Select the automation definition to edit.
5. Click *Change* to open the Automation Event Administration dialog box.
6. Make the necessary edits.
7. Click *Apply*.

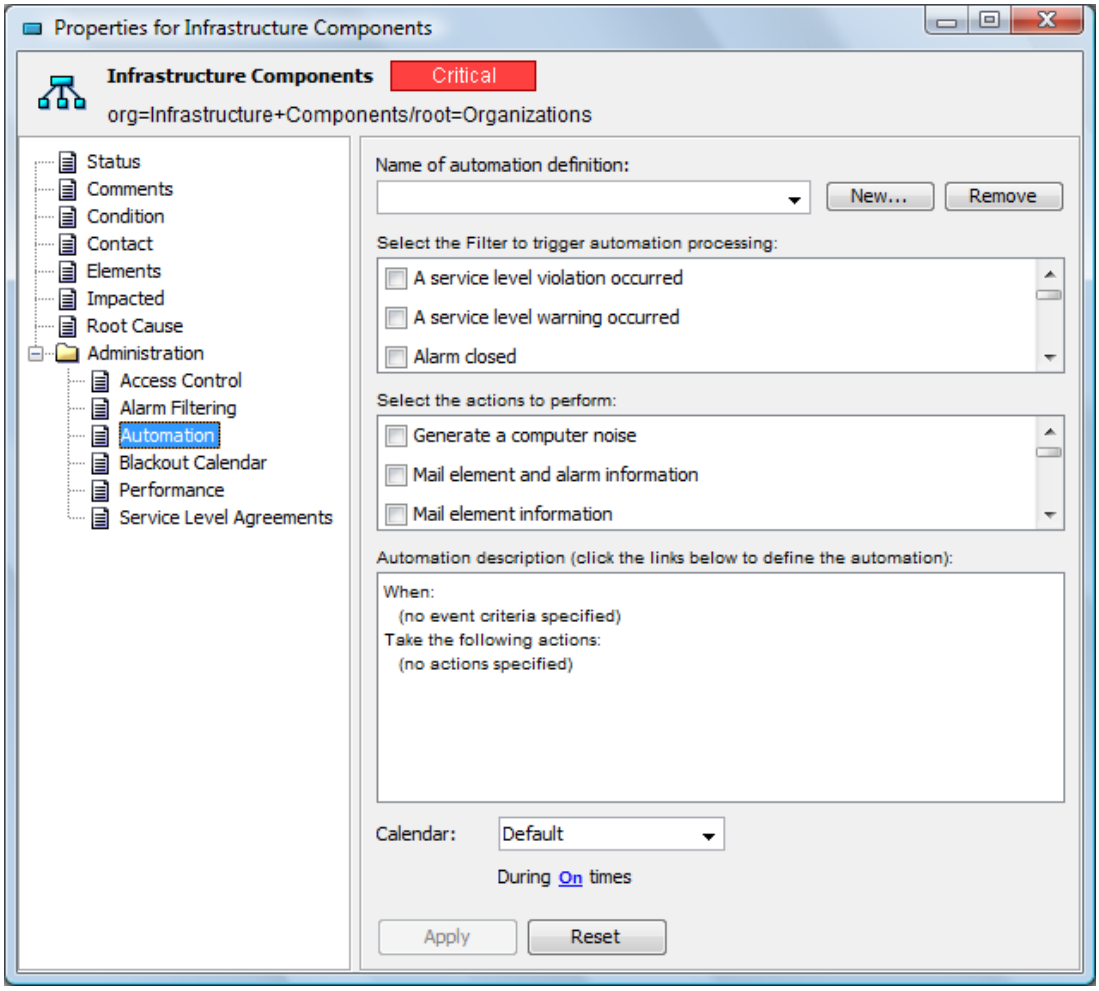
   The changes are saved for the automation event.

14.4 Defining Automation Events Directly on Elements

Automation events defined at the element-level are configured for the current user only. If an automation event needs to be defined for another user or group, it must be done through the *Automation* element under *Administration*.

To directly add an automation event:

1. In the *Explorer* pane, right-click the element for which an automation event should be created, then select *Properties* to open the *Status* property page.
2. In the left pane, under *Administration*, click *Automation* to open its property page:



3. Click *New*.
4. Specify a name for the automation event in the *Name of Automation Definition* field.
5. Select the check boxes for any of the filters to trigger the automation action in the *Select the Filter to Trigger Automation Processing* section.

   For more information, see Section 14.6, "Using Automation Filters," on page 226.
6. Select the check boxes for one or more actions to occur then the selected filter triggers the action from the *Select the Actions to Perform* section.

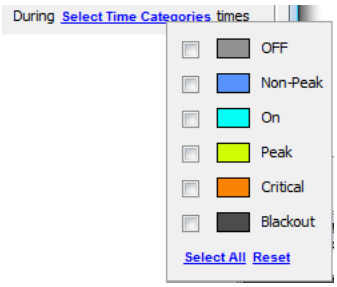
   For more information, see Section 14.7, "Using Automation Actions," on page 231.

7 In the *Automation Description* section, click a link to view, then select additional parameters for the automation event.

8 Click the *Calendar* drop-down list, then select the calendar to identify when to watch for the filter events.

For information about calendars, see Chapter 13, "Time Categories, Calendars, and Schedules," on page 197.

9 Click the *During XX Times* link, then select one or more check boxes to specify the applicable time categories within the selected calendar:



For more information about time categories, see Chapter 13, "Time Categories, Calendars, and Schedules," on page 197.

10 Click *Apply* to define the automation event for the element.

14.5 Monitoring and Managing Automation Events for an Element

Monitor the automation events associated with an element using the Properties dialog box.

- Section 14.5.1, "Monitoring Automation Events for an Element," on page 224
- Section 14.5.2, "Modifying Automations for Elements," on page 224

14.5.1 Monitoring Automation Events for an Element

To monitor an element's automation events:

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

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

If multiple automation definitions are defined for the element, click the *Name of Automation Definition* drop-down list, then select the automation.

14.5.2 Modifying Automations for Elements

A filter is assigned an automation definition that specifies the automated action taken when the filter triggers the action.

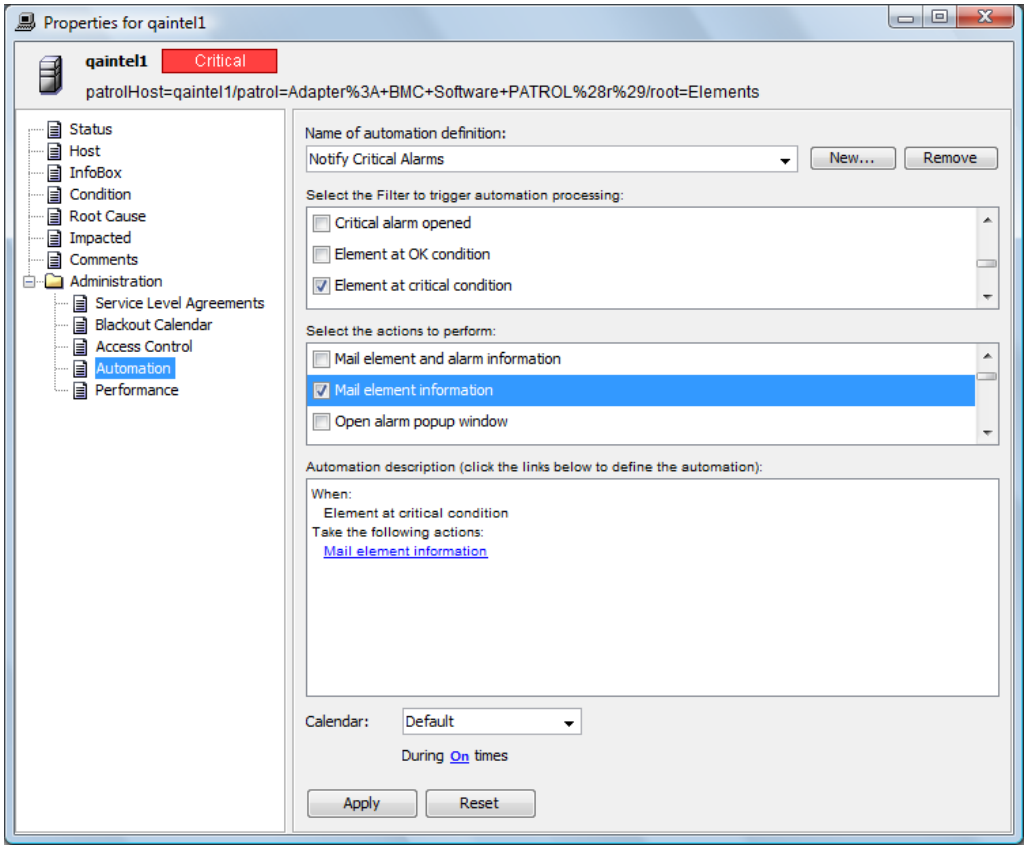
- "Modifying an Automation Definition" on page 225
- "Selecting a Specific Condition Level that Triggers the Automation Event" on page 226

Modifying an Automation Definition

To update an automation:

1. In the *Explorer* pane, navigate to the element with the automation definition.
2. Right-click the element, then select *Properties* to open the *Status* property page.
3. In the left pane, click *Automation* to open its property page.
4. Click the *Name of Automation Definition* drop-down list to select the automation definition.

   Automation settings display:



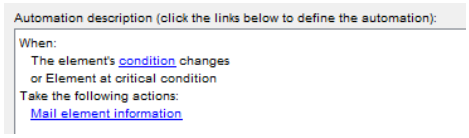
5. Make the necessary edits.
6. Click *Apply* to save changes.

Selecting a Specific Condition Level that Triggers the Automation Event

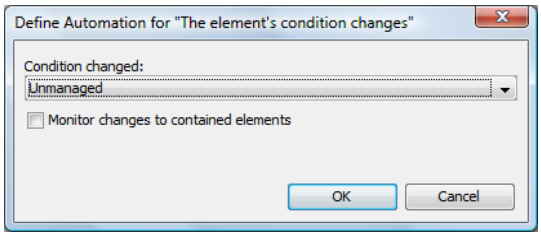
If *The Element's Condition Changes* is a filter, no condition level is defined by default, so the filter is triggered with any condition change.

To specify a condition level to trigger an automation event:

1 In the *Automation Description* section, click the condition link to open the Define Automation dialog box for setting the condition:



2 Click the *Condition Changed* drop-down list, then select a condition:



Select the element condition that determines when the filter is used.

For example, selecting *Unmanaged* triggers the action only when the element condition changes to Unmanaged.

3 Click *OK* to close the Define Automation dialog box.

14.6 Using Automation Filters

For each automation definition, a filters specifies an events that trigger actions when they occur. Operations Center supplies a set of default filters but new filters can also be defined. After defined, automation filters are available to select when creating automation definitions.

- Section 14.6.1, "Understanding the Default Filters," on page 226
- Section 14.6.2, "Filtering Alarm Events," on page 228
- Section 14.6.3, "Defining a New Filter," on page 229
- Section 14.6.4, "Modifying a Filter," on page 230
- Section 14.6.5, "Deleting a Filter," on page 230

14.6.1 Understanding the Default Filters

Any of the default definitions can be edited to meet specific requirements.

When more than one filter is selected for an automation event, any of the selected filters can be triggered for the automated action to occur. For example, if both the *Alarm Closed* and *Element Is at Major Condition* filters are selected, the specified action is triggered when either event occurs.

Alarm-related filters, such as any alarm opened, critical alarm opened, major alarm opened, and so on, trigger automation events when an alarm is added or updated. Note that alarm-based automations are not applied to elements that are configured to not show alarms.

Table 14-3 defines the default filters that can be used with any automation.

Table 14-3 *Default Automation Filters*

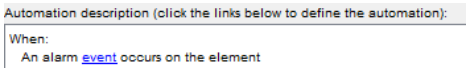
Filter	Description
A service level breach occurred	A Service Level breach alarm occurred.
A service level warning occurred	A Service Level warning alarm occurred.
An alarm is closed	An alarm of any type for the source element was closed.
An alarm event occurs on the element	By default, adding or updating a real-time alarm triggers this filter. You can change the alarm event type and alarm channel, as well as add other alarm information requirements. See the next section for details.
An alarm operation was performed	By default, any alarm operation triggers the automation event. To specify an operation, click the operation link in the *Automation Description* section and enter the command that should trigger the automation event.
Any alarm is opened	An alarm of any type occurred for the source element.
A critical alarm is opened	A critical alarm occurred for the source element.
The element is at an OK condition	A source element is in or changed to the OK state.
The element is at a critical condition	A source element is in or changed to critical alarm state.
The element is at an information condition	A source element is in or changed to information alarm state.
The element is at a major condition	A source element is in or changed to major alarm state.
Element is at a minor condition	A source element is in or changed to minor alarm state.
A major alarm is opened	A major alarm occurred for the source element.
A minor alarm is opened	A minor alarm occurred for the source element.
The element's condition changes	A source element's condition changes to and from any state. To specify a "changed to" condition, click the condition link in the *Automation Description* section, then select the condition that should trigger the automation event.

14.6.2 Filtering Alarm Events

NOTE: Alarm-based automations are not applied to elements that are configured to not show alarms.

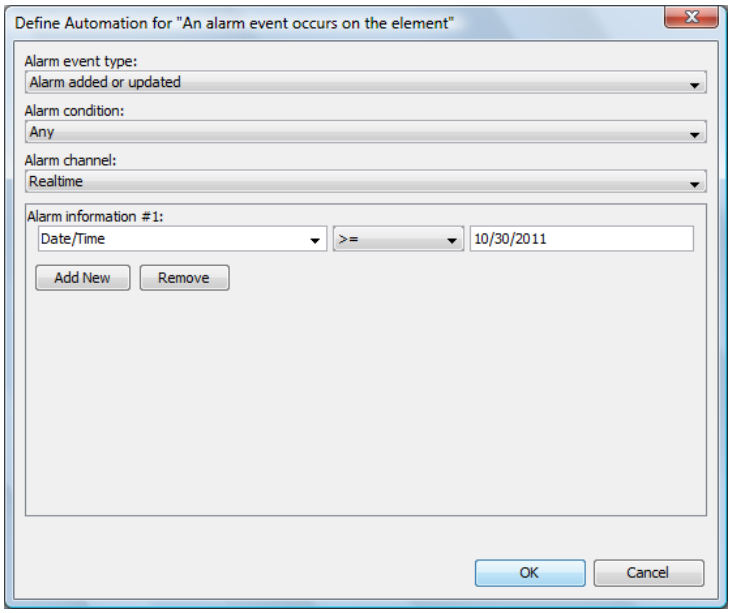
To define filters on alarm events:

1 When the selected filter is *An Alarm Event Occurs on the Element*, click the event link in the *Automation Description* section of the dialog box to select the alarm event, alarm channel, and other alarm information required to trigger the automation:



Click the event link to specify an alarm event that triggers the automation action.

2 Define alarm event requirements on this dialog box:



3 Select an alarm event type:

- *Alarm Added or Updated*
- *Alarm Added*
- *Alarm Updated*
- *Alarm Removed*

4 (Optional) Specify an alarm condition, alarm channel, and alarm information, such as a date range.

The automation event occurs only if all the alarm criteria are met.

5 To add alarm information criteria:

- In the first column, select a column name (such as *Severity*, *ID*, *Rule*, and so on).
- Incoming alarm data in the selected column is compared to the value entered in the far right column, using the selected comparison operator.

  For example, the previous figure shows alarms must have a date/time stamp equal to or greater than 12/10/2007.
- Click *Add New* to specify additional alarm information.

  Multiple criteria are evaluated using the "AND" operator.

14.6.3 Defining a New Filter

New automation filters are available to select when creating automation definitions.

To define a new filter:

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

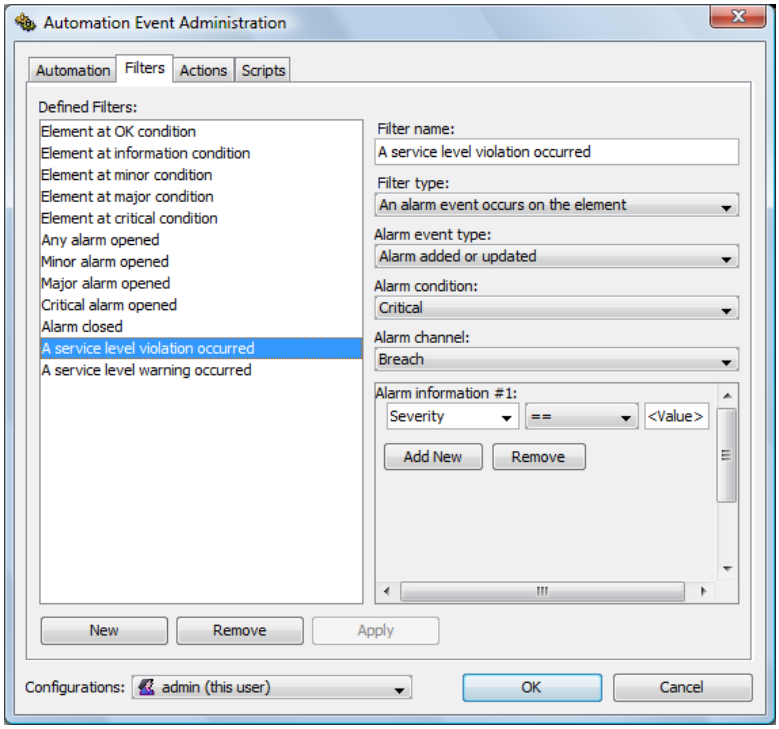
2 Right-click *Automation*, then select *Properties* to open the *Status* property page.

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

4 Click *Change* to open the Automation Event Administration dialog box.

5 Click the *Filters* tab.

6 Click the *Configurations* drop-down list, then select the user, group, or the Automation server for which the new filter is available:



7 Click *New*.

8 Specify a name for the new filter in the *Filter Name* field.

9 Click the *Filter Type* drop-down to select the type of filter, then do the following:

- If alarm-related, click the *Alarm Event Type*, *Alarm Condition*, and *Alarm Channel* drop-down lists, then select the filter parameters.

  For information about each parameter, see Section 14.6.1, "Understanding the Default Filters," on page 226.

- In the *Alarm Information #X* section, select the alarm column value to trigger the event.

  For more information, see Section 14.6.2, "Filtering Alarm Events," on page 228.

10 Click *Apply* to save the new filter.

14.6.4 Modifying a Filter

To update an automation filter:

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

2 Right-click *Automation*, then select *Properties* to open the *Status* property page.

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

4 Click *Change* to open the Automation Event Administration dialog box.

5 Click the *Filters* tab.

6 Click the *Configurations* drop-down list, then select the user, group, or the Automation server for which the filter is defined.

7 In the *Defined Filters* list, select a filter.

The filter's settings display on the right.

8 For some filters, there is an option to modify the parameters in the in the *Automation Description* section.

For more information about parameters, see Section 14.6.1, "Understanding the Default Filters," on page 226.

9 Click *Apply* to update the parameters.

14.6.5 Deleting a Filter

To delete an automation filter:

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

2 Right-click *Automation*, then select *Properties* to open the *Status* property page.

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

4 Click *Change* to open the Automation Event Administration dialog box.

5 Click the *Filters* tab.

6 Click the *Configurations* drop-down list, then select the user, group, or the Automation Server for which the filter is defined.

7 In the *Defined Filters* list, select the filter, then click *Remove* to delete the filter.

14.7 Using Automation Actions

For each automation definition, actions specify what happens when a filter is triggered. Table 14-4 lists the set of default actions supplied by the Operations Center.

Table 14-4 *Default Automation Actions*

Action	Parameters
Start the gong.	Plays an audio file, which can be configured.
Stop the gong.	Stops playing the audio file.
Start chirping.	Plays an audio file, which can be configured.
Stop chirping.	Stops playing the audio file.
Generate the computer noise.	Plays an audio file, which can be configured.

You can define new actions.

Some automation actions must execute on the server, while other automation actions can execute on the client or the server. The following automation actions must execute on the server:

- Mail element and alarm information
- Mail element information

To use automation actions:

- Section 14.7.1, "Understanding Action Parameters," on page 231
- Section 14.7.2, "Using the Open Alarm Pop-up Dialog Box Action," on page 233
- Section 14.7.3, "Defining a New Action," on page 233
- Section 14.7.4, "Modifying an Automation Action," on page 234
- Section 14.7.5, "Deleting an Automation Action," on page 235

14.7.1 Understanding Action Parameters

The *Actions* tab provides predefined actions that occur when triggered by a filter. You can define additional actions as needed.

Table 14-5 lists the different types of automation actions. Many automation actions require additional parameters, such as a sender and recipient e-mail address.

Table 14-5 *Types of Automation Actions*

Action	Parameters
Mail element and alarm information.	The sender e-mail address. The SMTP server address. The recipient e-mail address (optional, can use element).
Mail element information.	The sender e-mail address. The SMTP server address. The recipient e-mail address (optional, can use element).

Action	Parameters
Open alarm pop up window.	Alarm detail field #1 (optional). Specify an alarm column name to show in addition to the standard Operations Center alarm columns. Alarm detail field #2 (optional). Maximum alarms to keep in pop up. Default is 100 (optional). Alarm message expression (optional). Specify a regular expression to display a portion of the alarm description. If any text in the alarm description matches the regular expression, it is displayed with the alarm. Otherwise or if left blank, the entire alarm description is displayed. For example: ◆ /^.{20}/ Display the first twenty characters ◆ /http:.*/ Display everything after the first http: found ◆ /\d*/ Display the first number found
Page element.	The paging ID (optional, can use element). The paging message (optional, can use element). The SNPP server address (optional) The SNPP server port (optional).
Play an audio clip.	Name of audio file. Options: ◆ *Play Indefinitely* ◆ *Play Once* ◆ *Play for Duration (in Seconds)*
Post alarm to tec.	The hostname of the TEC server. The port of the TEC server (optional, defaults to `0`). The event class (optional, defaults to `LogFile_Base`). The event source (optional, defaults to `Formula`). The event severity (optional, defaults to `WARNING`).
Post to tec.	The hostname of the TEC server. The port of the TEC server (optional, defaults to `0`). The message to send to TEC The event class (optional, defaults to `LogFile_Base`). The event source (optional, defaults to `Formula`). The event severity (optional, defaults to `WARNING`).
Run a predefined script from the script library.	Library script name Script parameters and setup code. The script name must be unique. If two identical script names are in different directories, you must rename one of the scripts. Otherwise, errors occur when the script is run.
Run a script.	Script code. For more information about creating scripts and script selections, see the *Operations Center 5.5 Scripting Guide*.
Stop a running audio clip.	Name of audio file.

14.7.2 Using the Open Alarm Pop-up Dialog Box Action

The *Open Alarm Pop-up Dialog Box* action is valid for client automations only, meaning those configured for a user or group. It is invalid for automations on the Automation Server.

The *Open Alarm Pop-up Dialog Box* action is invalid for the following automation filters, because the *Open Alarm Pop-up Dialog Box* action explicitly assumes that it is displaying an alarm, as suggested by its name. The filters match events that are not produced directly by an alarm:

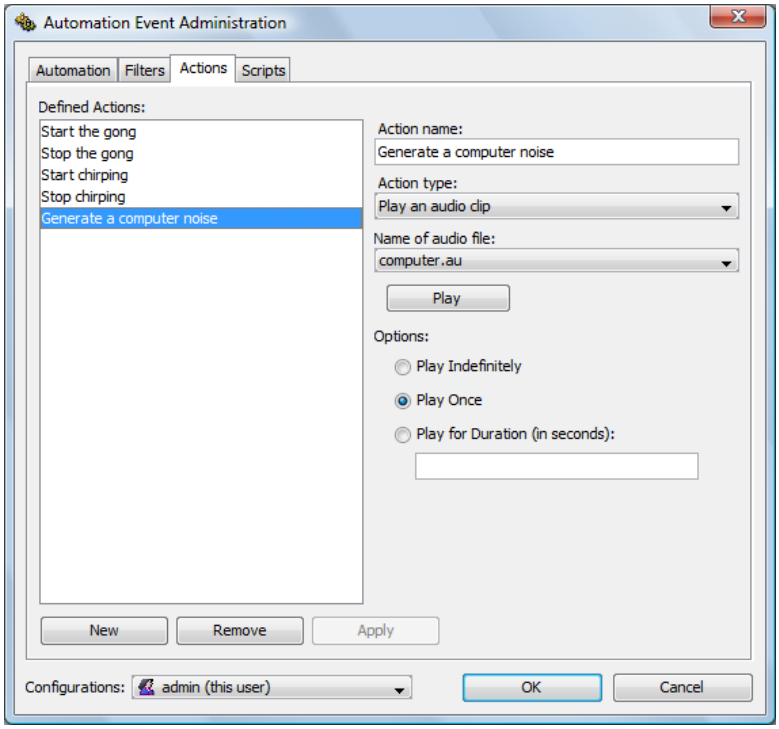
- Element at CRITICAL condition
- Element at MAJOR condition
- Element at MINOR condition
- Element at INFORMATION condition
- Element at OK condition
- The element's condition changes

14.7.3 Defining a New Action

New automation actions are available to select when creating automation definitions.

To define an automation action:

1. In the *Explorer* pane, expand *Administration*.
2. Right-click *Automation*, then select *Properties* to open the *Status* property page.
3. In the left pane, click *Automation* to open its property page.
4. Click *Change* to open the Automation Event Administration dialog box:



5. Click the *Actions* tab.

6 Click the *Configurations* drop-down list, then select the user, group, or the Automation server for which the action is available.

7 Click *New*.

8 Specify a name for the action in the *Action Name* field.

9 Click the *Action Type* drop-down list, then select an action to be performed when the event is triggered.

Depending on the action selected, additional drop-down lists and options lists display for selection.

10 Select the appropriate parameters for the action.

11 Click *Apply*.

The action displays in the *Defined Actions* section.

14.7.4 Modifying an Automation Action

It is possible to modify actions, such as changing the audio file or script associated with a particular action. You can also set a default behavior for updated automations.

- "Modifying an Automation Action" on page 234
- "Setting the Default Behavior for Updated Automations" on page 234

Modifying an Automation Action

To update an automation action:

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

2 Right-click *Automation*, and select *Properties* to open the *Status* property page.

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

4 Click *Change* to open the Automation Event Administration dialog box.

5 Click the *Actions* tab.

6 Click the *Configurations* drop-down list, and select the user, group, or *Automation Server* that the action is available for.

7 Select the action to modify.

8 Edit the action parameters.

9 Click *Apply* to save the changes.

Setting the Default Behavior for Updated Automations

By default, all automation actions that monitor changes in element conditions are performed when any of these automations is modified, even when no element condition change occurs. The default setting is:

```
Automations.element.change.never.fires=false
```

To modify this setting:

1 Open the *formula.properties* file.

2 Change the setting to True to prevent automatic execution of all automation actions related to element condition changes when any of them is updated.

14.7.5 Deleting an Automation Action

To delete an automation action:

1. In the *Explorer* pane, expand *Administration*.
2. Right-click *Automation*, and select *Properties* to open the *Status* property page.
3. In the left pane, click *Automation* to open its property page.
4. Click *Change* to open the Automation Event Administration dialog box.
5. Click the *Actions* tab.
6. Click the *Configurations* drop-down list, and select the user, group, or *Automation Server* for which the action is available.
7. Select an action, then click *Remove*.
8. Click *Apply* to delete the action.

15 Using Algorithms to Calculate Element State

By default, an element's state is calculated simply by inheriting the condition of it's most critical child element.

Optionally, alternate algorithms are used to apply a sequence of calculations to control state propagations and update with conditions that more accurately reflect the overall status of the Business Service View in which the element resides.

For example, if two Web servers support a Web store application, but only one is necessary to run it correctly, then an algorithm can be applied to the parent of the two servers allowing the state of the WebStore application to show as available as long as at least one Web server is running normally.

The following sections describe how to use algorithms to calculate element conditions:

- Section 15.1, "Understanding Algorithms," on page 237
- Section 15.2, "Setting an Algorithm for an Element," on page 239
- Section 15.3, "Verifying and Troubleshooting Algorithms Using the Algorithm Tracer," on page 240
- Section 15.4, "Modifying the Algorithm Library," on page 242
- Section 15.5, "Algorithm XML Tags Reference," on page 243

15.1 Understanding Algorithms

An algorithm is a set of rules used to calculate the condition of an element. Sometimes the algorithm is more complex and consists of multiple rules: an initial element set (any elements to consider), the intermediate state of the calculation (such as a filtered set of elements on which to operate), and the result of the calculation (ElementCondition).

Algorithms can be applied to natively defined elements in the *Services* hierarchy, and for elements of event-based adapters in the *Elements* hierarchy. They are not available for elements created using object-based adapters, such as NetIQ, OpenView, InterCommunication, and Spectrum.

The following sections describe the default algorithms library as well as custom examples that can be used as-is or further modified:

- Section 15.1.1, "Default Algorithm Types," on page 238
- Section 15.1.2, "Custom Algorithm Examples," on page 238

15.1.1 Default Algorithm Types

By default, the following algorithm types are available:

- **average:** Calculates the average condition of all children.
- **bands:** Uses a series of thresholds to sets condition based on the percentage of children that are CRITICAL:
  - When 25%-50% of children are CRITICAL, sets element condition to MINOR.
  - When 50%-75% of children are CRITICAL, sets element condition to MAJOR.
  - When 75%-100% of children are CRITICAL, sets element condition to CRITICAL.
  - If less than 25% of children are CRITICAL, sets condition to highest condition occurring in at least 25% of children.
- **bemEndUserBand:** Sets condition based on thresholds against response times on BEM end user tests. Values are the same as BEM defaults.
- **bemSyntheticBand:** Sets condition based on thresholds for response times on BEM synthetic tests.
- **count:** Looks for any condition occurring for at least 50% of children.
- **highest:** Takes the highest condition from all children.
- **lowest:** Takes the lowest condition from all children.
- **paramCount:** Looks for any condition occurring in a specified percentage of children. Specify percentage threshold, default condition, and description.
- **paramSet:** Sets the condition. Specify description and resulting condition.
- **paramHighest:** Takes the highest condition. Specify default condition, child type, and description.
- **paramBand:** Sets two thresholds to test for percentage of children with selected condition. Specify two percentage thresholds, test conditions, and resulting conditions.
- **paramScript:** Runs a script. Specify script code or name of script located in `/OperationsCenter_install_path/database/scripts`.
- **paramReduce:** Takes the highest condition of children with matching element property value. Specify invert value, property name, and value.
- **paramElementPropertyBand:** Sets condition based on thresholds against selected property value. Specify property name, upper and lower thresholds, description, and resulting condition.
- **paramBemSyntheticBand:** A parameter based version of the bemSyntheticBand algorithm. Specify upper and lower thresholds for all condition levels.
- **suppressSensitive:** If all children (NAM and ORG) are suppressed, then the element is suppressed. Otherwise, it uses the highest condition.

15.1.2 Custom Algorithm Examples

The algorithm library includes various examples that can be used as is, customized to suit your requirements, or just referred to as you create new algorithms.

For more information about creating new algorithms, see Section 15.4, “Modifying the Algorithm Library,” on page 242.

The example algorithms include the following:

- **ciscorouters:** Uses a reduction and a regular expression to look at the condition of all children with class `cisco` and takes the highest condition.
- **hosts:** Uses a reduction to look at the condition of all children with class `host` and calculates the average.
- **invoking:** Finds all children and then uses the `invoke` tag to call the count algorithm.
- **ports:** Uses a script fragment to look at the condition of all children with class `port` and takes the highest condition.
- **splitting:** Looks at the condition of all children with class router, uses `split` and `branch` tags to check to see the percentage of CRITICAL children, then uses `band` tags to set the condition.

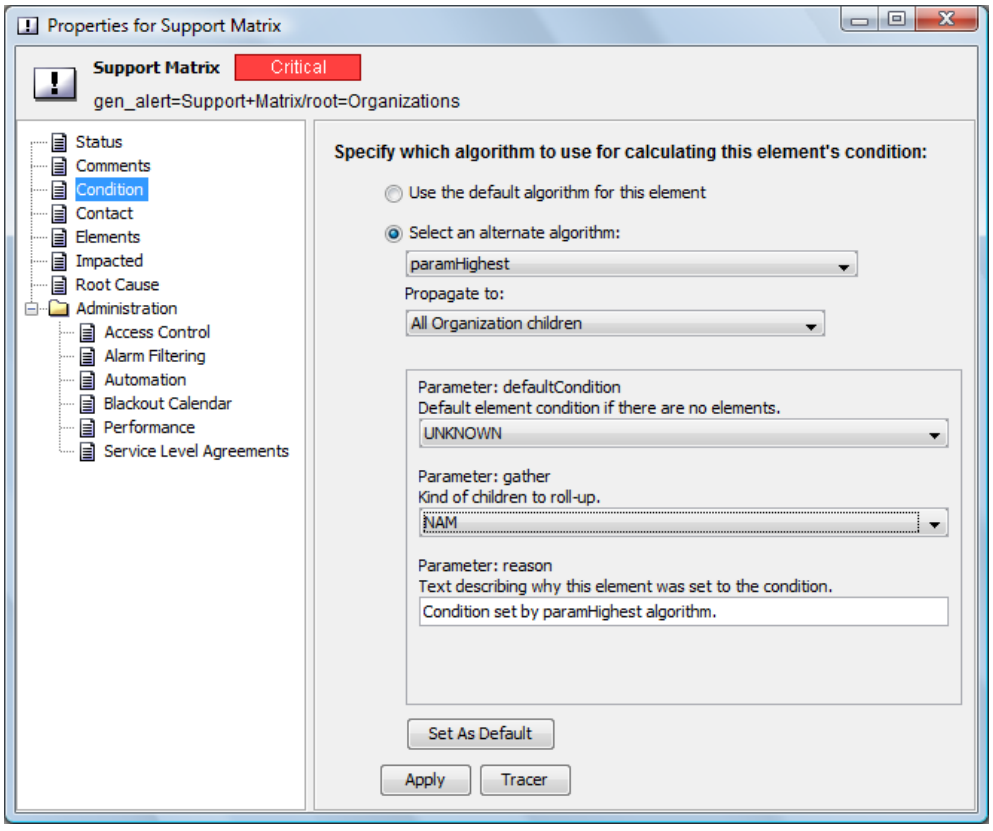
15.2 Setting an Algorithm for an Element

Algorithm calculations are available for event-based adapters (found in the *Elements* hierarchy), and for any elements defined natively to the *Service Models* and *Locations* hierarchies. Algorithm calculations are defined in the element's Properties dialog box using the *Condition* tab.

To set up an algorithm to calculate an element's condition:

1 In the *Explorer* pane, right-click the element, then select *Properties*.

2 In the left pane, click *Condition*.



3 Choose *Select an Alternate Algorithm*, and do the following:

3a Select the algorithm type.

For a description of each algorithm type, see Section 15.1.1, "Default Algorithm Types," on page 238 and Section 15.1.2, "Custom Algorithm Examples," on page 238.

3b Select from one of the propagation rules:

Propogate to No Children: Apply calculations to the current element only.

Propogate to All children: Apply calculations to the current element and all of the element's children.

Propogate to All children, Except Those with No Children (Might Force Discovery):

Apply calculations to the current element and only those children that have native children (element children for *Elements*, and organization children for *Service Models* and *Locations*). Operations Center may launch a discovery process to determine if the child element has children, if not already discovered.

Propogate to All children, Except Those with No Discovered Children (Discovery Is Not Forced): Apply calculations to the current element and only those children that are currently known to have native children (element children for *Elements*, and children for *Locations* and *Service Models*). A discovery process is NOT undertaken to determine the child element's status.

3c Define other parameters as required for the custom algorithm.

When available, customize the reason text that displays a message in the *Notes* column of the *Summary* view or in element tool tips. For example, if a `band` command fires because 50% of the objects are CRITICAL, the notes message might display `Half the routers are critical` as the reason for the firing.

4 Click *Tracer* to verify the algorithm achieves the desired results.

For more information about using the Algorithm Tracer, see Section 15.3, "Verifying and Troubleshooting Algorithms Using the Algorithm Tracer," on page 240

15.3 Verifying and Troubleshooting Algorithms Using the Algorithm Tracer

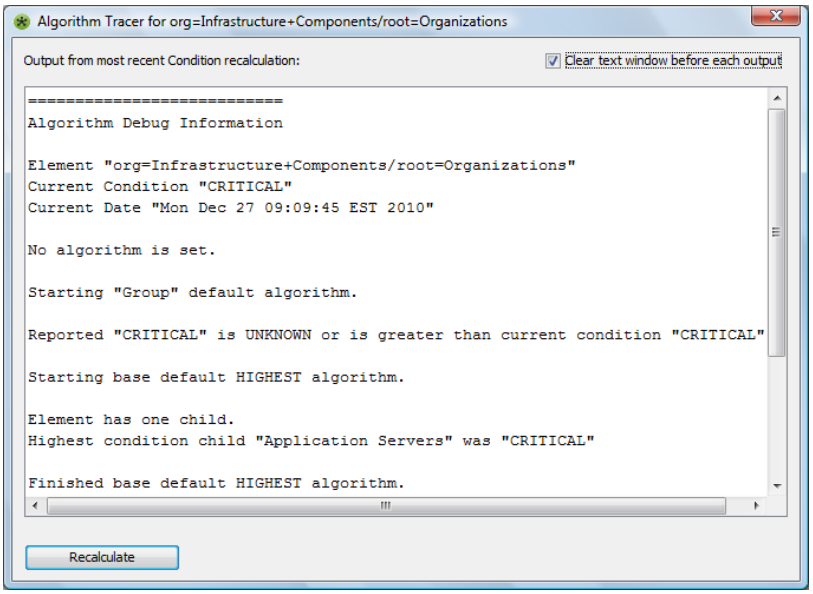
The Algorithm Tracer is a tool that displays the most recent algorithm trace information for an element and identifies the reason for its current condition.

Use the Algorithm Tracer to:

- Trace and validate how an element's condition is determined
- Verify that condition was calculated as intended
- See the affect on the state of related elements immediately after algorithm modifications
- See how a change in condition affects the parent elements

In the following illustration of the Algorithm Tracer, the element has five children, all with UNKNOWN condition, so the UNKNOWN condition is propagated.

Figure 15-1 *Algorithm Tracer Showing Element Condition Calculations*



The following sections provide instructions on using the Algorithm Tracer:

- Section 15.3.1, “Tracing and Validating the Algorithm Calculation,” on page 241
- Section 15.3.2, “Testing an Algorithm’s Effect on Parent Condition,” on page 242

15.3.1 Tracing and Validating the Algorithm Calculation

To trace the algorithm calculation:

1. In the *Explorer* pane, right-click an element, then select *Properties*.
2. In the left pane, click *Condition*.
3. If necessary, specify the alternate algorithm.

   For more information, see Section 15.2, “Setting an Algorithm for an Element,” on page 239.
4. Click *Tracer* to open the Algorithm Tracer.
5. To perform an algorithm recalculation immediately, click *Recalculate*.

   Leave the Algorithm Tracer dialog box open while modifying the *Condition* tab.
6. If you select a different algorithm or modify a parameter, click *Recalculate* to recalculate the condition immediately and view results.

   By default, only the output for the most recent algorithm recalculation is displayed. To view all prior outputs as well as the most recent output, deselect the *Clear Text Window Before Each Output* option.

15.3.2 Testing an Algorithm's Effect on Parent Condition

To test how parent condition is affected by an algorithm:

1. Set a condition algorithm for the parent element with propagation set to include all children. Click *Tracer* to open the Algorithm Tracer and click *Recalculate*.
2. Set the condition algorithm for the child element. Click *Tracer* to open the Algorithm Tracer and click *Recalculate*.
3. View any changes to the parent element's condition.

15.4 Modifying the Algorithm Library

All algorithms surfaced by the algorithm library are algorithms that are defined in the `/OperationsCenter_install_path/database/examples/Algorithms.xml` file.

Make a copy of the `Algorithms.xml` file to customize existing algorithm definitions or to create additional algorithms. While any XML editor can be used to edit the `Algorithms.xml` file, the Algorithms Editor makes changes to the algorithm library easy.

The `/OperationsCenter_install_path/database/examples/Algorithms_examples.xml` file contains additional commented examples on how algorithms can be implemented. These examples are untested to a production scale and should be implemented carefully, one at a time, copying only those you expect to use into the active Algorithms.xml file.

When creating custom algorithms, keep in mind:

- If a deleted algorithm is used by an element, the element's algorithm is reset to the default algorithm.
- Do not set more than 10 banding thresholds.

To customize the Algorithm Library by editing the `Algorithms.xml` file:

1. Copy the `/OperationsCenter_install_path/database/examples/Algorithms.xml` file into the `/OperationsCenter_install_path/database` directory.

   If this step is not performed, manual changes in the default `Algorithms.xml` file are lost Operations Center software is reinstalled or upgraded.
2. In the *Explorer* pane, expand *Administration* > *Server*.
3. Right-click the *Algorithms* element and select *Edit Algorithm Definition* to open the Algorithm Editor.

4 Edit `algorithm` tags or create new definitions as required.

For general information about using the Algorithm Editor, see Appendix A, "The Operations Center XML Editor," on page 253.

For information about specific tags, see Section 15.5, "Algorithm XML Tags Reference," on page 243.

It is not necessary to restart the Operations Center server for changes to be applied.

15.5 Algorithm XML Tags Reference

An algorithm consists of a series of statement to execute and calculate the state of objects. This section describes the basic XML tags used in defining algorithms.

- Section 15.5.1, "Understanding the Algorithms.xml File," on page 243
- Section 15.5.2, "Understanding the Top-Level Tags," on page 244
- Section 15.5.3, "Understanding timebasedbranch and timebasedsplit," on page 250
- Section 15.5.4, "Understanding Parameter Evaluation," on page 251

15.5.1 Understanding the Algorithms.xml File

The `Algorithms.xml` file is formatted in XML which executes from the top down through a set of elements, the state calculation passes from statement to statement until there are no more statements, or a statement terminates the calculation.

Within an algorithm definition, many types of actions can be performed including:

- **Gather elements:** Determines which group of element children to look at when calculating condition.

  For more information, see `<gather>` in Table 15-2 on page 245.
- **Eliminate some elements:** Filters out any elements that should not affect the calculation.

  For more information see `<conditionreduce>`, `<matchreduce>`, and `<scriptreduce>` in Table 15-2 on page 245.
- **Simple calculations:** Performs basic arithmetic operations, such as highest, average, or lowest.

  For more information, see `<average>`, `<highest>`, and `<lowest>` in Table 15-2 on page 245.
- **Fire a script:** Initiates a script that calculates the condition.

  For more information, see `<script>` in Table 15-2 on page 245.
- **Banding Thresholds:** Compares the results of arithmetic operations to a defined threshold or a set up to 10 high and low limits to calculate the element's condition.

  For more information, see `<band>` and `<elementpropertyband>` in Table 15-2 on page 245.
- **Split command:** Allows the original condition value to be passed to various branches of the algorithm to be modified as necessary without affecting other algorithm calculations.

  For more information, see `<split>` and `<timebasedsplit>` in Table 15-2 on page 245.

The usual sequence of an algorithm definition is to gather, reduce, then perform a condition calculation, often by highest or by count. A `skip` tag can be used break from current calculations and continue using a different algorithm sequence if certain conditions are met.

The following excerpt from the `Algorithms.xml` file shows the file header and the bandPercent algorithm definition. Excluding any children with the state of UNKNOWN or UNMANAGED, it uses a series of thresholds to set condition based on a percentage of children having a specified condition. For example, if 75% or more children are MAJOR, the resulting calculation is CRITICAL.

```
<?xml version="1.0"?>
<!DOCTYPE algorithms PUBLIC
 "-//Managed Object Solutions, Inc.//DTD algorithms 1.0//EN"
  "http://www.ManagedObjects.com/dtds/Algorithms.dtd>

<algorithms>
   <algorithm name="_bandPercent">
      <gather relationship="NAM" />
      <gather relationship="ORG" />
      <conditionreduce testCondition="UNKNOWN" />
      <conditionreduce testCondition="UNMANAGED" />
      <band testCondition="CRITICAL" amount="75%" result="CRITICAL" />
      <band testCondition="MAJOR" amount="75%" result="CRITICAL" />
      <band testCondition="CRITICAL" amount="50%" result="MAJOR" />
      <band testCondition="MAJOR" amount="50%" result="MAJOR" />
      <band testCondition="CRITICAL" amount="25%" result="MINOR" />
      <band testCondition="MAJOR" amount="25%" result="MINOR" />
      <band testCondition="MINOR" amount="50%" result="MINOR" />
      <band testCondition="OK" amount="100%" result="OK" />
      <set result="INFO" />
   </algorithm>
</algorithms>
```

It is important that the XML be well-formed. Options for an algorithm are specified between the `<algorithm></algorithm>` tags. For information about using specific tags, continue to Section 15.5.2, "Understanding the Top-Level Tags," on page 244.

15.5.2 Understanding the Top-Level Tags

Each `Algorithms.xml` file requires one `algorithms` tag to be declared. Within the `algorithms` tag, `algorithm` and `defineCommand` tags are used to define each algorithm.

Table 15-1 lists the top-level tags for the `Algorithms.xml` file.

Table 15-1 *Top-Level Algorithm Tags*

Tag	Description
`<algorithms>`	Declares a series of named algorithms.Place a series of `algorithm` tags inside to define different algorithm definitions, or use a series of `defineCommand` tags to define the algorithm calculations.
`<algorithm>`	Defines a new algorithm. Specify the following attributes as required: ◆ **name:** Identifier for the algorithm. Displays when selecting an alternate algorithm for an element. Can contain any number of `<exec>` and `<split>` declarations.

Tag	Description
<defineCommand>	Used in lieu of the algorithm tag, declares a new command using exec tags. A Java class must be available that adheres to the com.mosol.util.algo.Algorithm interface. Specify the following attributes as required: ◆ **name:** Name of the algorithm. ◆ **class:** Class of the algorithm. The engine executes Class.forName on the class specified and creates new instances with a no-argument constructor. The class has its initializeAlgorithm method called when instantiated. Parameter substitution can be used within definecommand statements.

Inside each algorithm and defineCommand tag, the tags listed in Table 15-2 are used to setup the specific statements and calculations that allow the algorithm to compute condition for an element.

Table 15-2 *Algorithm Definition Tags*

Tag	Description
<average>	Returns average condition of the elements in the active set. Specify the following attributes as required: ◆ **defaultCondition:** Condition to be set if there are no gathered elements. Specify INITIAL (Operations Center reports this as UNMANAGED), UNKNOWN, CRITICAL, MAJOR, MINOR, INFORMATIONAL, or OK. ◆ **reason:** Descriptive text explaining why the condition was set. ◆ **rounding:** If lower, rounds the calculation to the more severe condition value. Specify lower or higher. If not specified, the less severe condition level is returned.
<band>	A threshold evaluation used to set the condition based on a percentage of elements which a specific condition. Successive band tags are commonly used to assign declining levels of severity to certain numbers of elements at a condition level. Once a threshold band is matched, the resulting condition is set, and the computation stops. Specify the following attributes as required: ◆ **testCondition:** Condition to be tested. Specify INITIAL (Operations Center reports this as UNMANAGED), UNKNOWN, CRITICAL, MAJOR, MINOR, INFORMATIONAL, or OK. ◆ **amount:** Matching criteria as a number or a percentage. ◆ **result:** Condition to be set if there is there is a match. Specify INITIAL (Operations Center reports this as UNMANAGED), UNKNOWN, CRITICAL, MAJOR, MINOR, INFORMATIONAL, or OK. ◆ **reason:** Descriptive text containing the reason why the condition was set. Used for root cause information and element notes. ◆ **skip:** Name of algorithm to skip to if the algorithm passes. If skip is not defined, the next tag statement found is evaluated. ◆ **invert:** If yes, inverts the algorithm's test criteria.

Tag	Description
<branch>	Branches are used within a split tag to apply a switch case or “if then, else” logic.Define branches to reduce a set of objects by class, name or condition; or evaluate condition on a single object or a class of objects. Use this strategy when service model's dependencies are less organized and there are multiple critical dependencies that need to be evaluated. Evaluate the conditions which would cause the focus of the algorithm to be most severe first, and then evaluate for less critical results. The branch tag can contain any other tags except the algorithms and algorithm tags.
<conditionreduce>	Excludes an element from the gathered elements if it is of a specified condition. Specify the following attributes as required: ◆ **testCondition:** Condition to be tested. Specify INITIAL (Operations Center reports this as UNMANAGED), UNKNOWN, CRITICAL, MAJOR, MINOR, INFORMATIONAL, or OK. ◆ **invert:** If yes, inverts the algorithm’s test criteria and elements remain as a gathered elements if not the specified condition.
<count>	Looks for a condition with the given number of elements. The count tag can be set to 50% to try to find, in descending order of severity, a triggering number of events. If at least 50% of elements are not of a severity above OK, it does not fire. Specify the following attributes as required: ◆ **amount:** Matching criteria as a number or a percentage. ◆ **defaultCondition:** Condition to be set if there are no gathered elements. Specify INITIAL (Operations Center reports this as UNMANAGED), UNKNOWN, CRITICAL, MAJOR, MINOR, INFORMATIONAL, or OK. ◆ **reason:** Descriptive text explaining why the condition was set.
<elementpropertyband>	Sets a condition to result if a specified element property has a value within the upper and lower points defined in the algorithm. The intent of this algorithm is to be used with the Experience Manager Suite to set thresholds of integer-based element properties, or namely response times. If the value of the given element property falls between the upper and lower bound defined, the result condition is set to the element that this algorithm is set to. If not, the conditionState is not modified. In the case of the Operations Center Experience Manager, the min and max values are used in milliseconds. Specify the following attributes as required: ◆ **elementProperty:** Name of the element property to evaluate. ◆ **min:** Lower bound threshold. Specify a number in ms. ◆ **max:** Upper bound threshold. Specify a number in ms. ◆ **result:** Condition to be set if the elementProperty value fall within the min and max values. ◆ **reason:** Descriptive text containing the reason why the condition was set. Used for root cause information. ◆ **skip:** Name of the algorithm to skip if the algorithm passes.

Tag	Description
<exec>	The exec tag has been superseded with separate XML tags. As a result, the other Algorithm tags can be used without the exec tag while providing better XML validation. Use the exec tag only for backwards compatibility for existing algorithms.xml files.
<gather>	Determines the set of elements to perform the calculation against based upon the element's relationship to the current element. Multiple gather tags can be used to more than one type of relationship. Specify the following attributes as required: ◆ **relationship:** Type of element relationship to gather. Specify ORG for elements in the *Service Models* hierarchy, or NAM for elements in the *Elements* tree.
<highest>	Takes the most severe condition of all elements. This corresponds to the default behavior if no algorithm is set for a given element. Specify the following attributes as required: ◆ **defaultCondition:** Condition to be set if there are no gathered elements. Specify INITIAL (Operations Center reports this as UNMANAGED), UNKNOWN, CRITICAL, MAJOR, MINOR, INFORMATIONAL, or OK. ◆ **reason:** Descriptive text containing the reason why the condition was set. Used for root cause information.
<invoke>	Permits algorithms to be called. This is the equivalent of invoking a subroutine. Specify the following attributes as required: ◆ **name:** Name of algorithm called. If the invoke tag is used to invoke another algorithm that is parameterized, the parameters must appear with the same name as the parameters in the top-level algorithm. Otherwise, the parameters values cannot be set.
<lowest>	Takes the least severe condition of all elements. Specify the following attributes as required: ◆ **defaultCondition:** Condition to be set if there are no gathered elements. Specify INITIAL (Operations Center reports this as UNMANAGED), UNKNOWN, CRITICAL, MAJOR, MINOR, INFORMATIONAL, or OK. ◆ **reason:** Descriptive text containing the reason why the condition was set. Used for root cause information.

Tag	Description
<matchreduce>	Removes an element from the calculation based on a regular expression match against a property value. Specify the following attributes as required: ◆ **property:** Name of the property to match. For example, for matching against: ◆ element dName, use property=dname ◆ element name, use property=name ◆ element condition, use property=condition ◆ an element property, use property=*element_property_name* ◆ an element's class, use property=class ◆ **value:** Regular expression to match against the property's value. Use an "i" at the end for case insensitivity. ◆ **invert:** If yes, inverts the meaning of the algorithm's test criteria.
<parameter>	Modify behavior or values in the algorithm. Parameter values can be edited by the user when the algorithm type is selected for the element. Specify the following attributes as required: ◆ **name:** Parameter name. ◆ **description:** Descriptive text containing the reason why the condition was set. Used for root cause information. ◆ **type:** Type of parameter to identify acceptable values. Specify one of the following: text — Arbitrary text string. condition — INITIAL (Operations Center reports this as UNMANAGED), UNKNOWN, CRITICAL, MAJOR, MINOR, INFORMATIONAL, OK. boolean — YES, NO property — name, dname, className, marker kind — NAM, ORG, TOP, MAP choice — A specific set of pipe-delimited (|) choices. For example, enter: 00%|75%|50%|25% ◆ **default:** Default value based on selected type. Parameter values are stored in the Configuration Storage database for each associated element. Care should be taken when adding, removing, or changing algorithm parameters for an algorithm that in use.

Tag	Description

`<script>`	Invokes FormulaScript code. The script can examine the context of the condition calculation, extend or reduce the set of elements, set the state of the computation, change the result, or generally update the element's state in any way. Specify the following attributes as required:
	◆ **script:** A FormulaScript to be evaluated. Can be a small amount of code or a direct reference to any file located in the `/`*`OperationsCenter_install_path`*`/database/scripts` directory.
	◆ **resolveScriptParameters:** If `true`, the script has parameters that need to be resolved. Defaults to `false`. Parameters allow substitution of text that can differ per element. Be aware that using the braces ("{" and "}") in the script might cause conflicts with parameter substitution, because these characters signify substitution.
	Information that can be passed while using the script tag:
	◆ **subject:** The element whose condition is being calculated.
	◆ **server:** The Operations Center server.
	◆ **adapter:** The adapter from which the element originates.
	◆ **conditionState:** The current condition state. Inside the script, these tasks can be performed:
	◆ Finish the algorithm: `conditionState.setState(State.FINISHED)`
	◆ Reduce the elements: `conditionState.getElements().remove(elem)`
	◆ Iteratively review the current set of elements:
	`for( var enum = conditionState.getElements().elements(); enum.hasMoreElements();`
	`// Do something with element writeln( enum.nextElement() ) The return value of the script is ignored.`
	For more information about creating scripts and scripts available in the Script Library, see the *Operations Center 5.5 Scripting Guide*.

`<scriptreduce>`	Removes an element passed from the gathered elements based on a script return code of type `java.lang.Boolean`. True to remove an element and False to leave an element in the gathered set. Specify the following attributes as required: ◆ **script:** Script to be evaluated. Can be a small code snippet or a direct reference to any file located in the `/`*`OperationsCenter_install_path`*`/database/scripts` directory. ◆ **resolveScriptParameters:** If True, the script has parameters that need to be resolved. Defaults to False. Parameters allow substitution of text that can differ per element. Be aware that using the braces ("{" and "}") in the script might cause conflicts with parameter substitution, because those characters signify substitution. ◆ **invert:** If yes, inverts the return code meaning so that True leaves an element and False removes it.

Tag	Description
	Information that can be passed while using the `scriptreduce` tag: ◆ **subject:** The element whose condition is being calculated. ◆ **server:** The Operations Center server. ◆ **adapter:** The adapter from which the element originates. ◆ **element:** The element to test for removal. ◆ **state:** The current condition. For more information about creating scripts and scripts available in the Script Library, see the *Operations Center 5.5 Scripting Guide*.
`<set>`	Sets an arbitrary condition and finishes the calculation immediately. Specify the following attributes as required: ◆ **result:** Condition to be set. ◆ **reason:** Descriptive text containing the reason why the condition was set. Used for root cause information.
`<split>`	Creates copies of the original condition and passes it to each `branch` tag defined within the `split` tag. This allows each branch to perform calculations with the same state information without affecting each other.
`<timebasedbranch>`	Enables branching based on calendars and time categories. Uses the `calendar` and `timecategories` parameters. When selecting the algorithm, the user specifies a calendar and time categories. For more information, see Section 15.5.3, "Understanding timebasedbranch and timebasedsplit," on page 250.
`<timebasedsplit>`	Functions the same as the `split` tag based on calendars and time categories. For more information, see Section 15.5.3, "Understanding timebasedbranch and timebasedsplit," on page 250.

15.5.3 Understanding timebasedbranch and timebasedsplit

The `<timebasedbranch>` and `<time-basedsplit>` tags enable an algorithm to behave differently, depending on the day and time. One simple application is to set the condition based on a specific time category.

All algorithms are state-driven and algorithm logic fires only when a state change occurs, or when a recalculation occurs. So when using calendar-based algorithms, it is important to remember that this type of algorithm is not reevaluated when a calendar is edited, or when a calendar switches from one time category to another. For example, you might expect the condition to change when the calendar goes from Peak to Off-peak at 5 PM. However, this change does not fire a recalculation at 5 PM, it is only recalculated when a state change occurs.

The following example sets the highest condition during the Peak and Critical time categories. The algorithm gathers everything on ORG and NAM, then looks for any matches of 2 or 3 for *AlgorithmCalendar*. If a match is found, the normal highest state is used. The algorithm sets a MINOR condition for all time periods other than these two time categories.

```
  <algorithm name="calendarSplitting">
        <exec command="gather" relationship="NAM" />
        <exec command="gather" relationship="ORG" />
            <timebasedsplit>
                <timebasedbranch calendar="AlgorithmCalendar"
timecategories="2|3">
                    <exec command="highest" reason="Condition set by 'exec'
statement using 'highest' on match of calendar time categories in algorithm." /
>
                </timebasedbranch>
                <timebasedbranch>
                    <set result="MINOR" reason="Condition set by 'set' statement
for NOT matching calendar time categories in algorithm." />
                </timebasedbranch>
            </timebasedsplit>
    </algorithm>
```

Where, the `AlgorithmCalendar` is the name of the calendar in Operations Center. The `timecategories` parameter is a pipe-delimited (`|`) list of time category IDs and uses the following numbers for the `timecategories` parameter: `BLACKOUT = 0`, `CRITICAL = 1`, `PEAK = 2`, `ON = 3`, `NONPEAK = 4`, and `OFF = 5`.

15.5.4 Understanding Parameter Evaluation

The following information should be considered when using parameters with the tags described in Table 15-2.

- The initializeAlgorithm() method, invoked when the `algorithms.xml` file is loaded, stores initial attribute values without converting them to their base types.
- The text string given as an attribute (such as "{condition}," "MAJOR," and so on) is not converted to its base type (ElementCondition) until the algorithm statement is evaluated against a specific element after parameter substitution takes place. An invalid value (`result="NO"`) does not log an exception until algorithm evaluation time, not upon initial load of the `algorithms.xml` file.
- The actOnState() method, invoked when the algorithm statement is evaluated for a specific element, performs parameter substitution for each attribute value using the `resolveParameters()` method in the ConditionState class. A code fragment might be:

  ```
  String condition =
  ((com.mosol.Formula.Server.algo.ConditionState)state).resolveParameters(condit
  ionString);
  ```

  where *state* is the condition as passed in as a parameter to `actOnState()`, and *conditionString* is the attribute value stored in the `initializeAlgorithm()` method.

A The Operations Center XML Editor

The Operations Center XML Editor provides a user interface to create and edit XML files. Menu options and click+drag actions can be used to add, remove, and rearrange *XML* elements. Viewing elements and attributes in a tree structure without the coding characters used in the XML raw text format makes it easier to work with the *XML* elements in the XML Editor.

- Section A.1, "Understanding and Accessing the XML Editor," on page 253
- Section A.2, "Managing XML Files," on page 254
- Section A.3, "Adding Elements," on page 257
- Section A.4, "Adding and Editing Attributes," on page 259
- Section A.5, "Moving Tags," on page 259
- Section A.6, "Deleting Tags," on page 260

A.1 Understanding and Accessing the XML Editor

The XML Editor provides a user interface for creating and editing XML files. Menu options and click+drag actions can be used to add, remove, and rearrange *XML* elements. Viewing elements and attributes in a tree structure without the coding characters used in the XML raw text format makes it easier to work with the *XML* elements in the XML Editor.

The XML Editor can be used to edit adapter hierarchy files and is available through the Operations Center Console. XML files are commonly used to:

- Edit adapter hierarchy files.

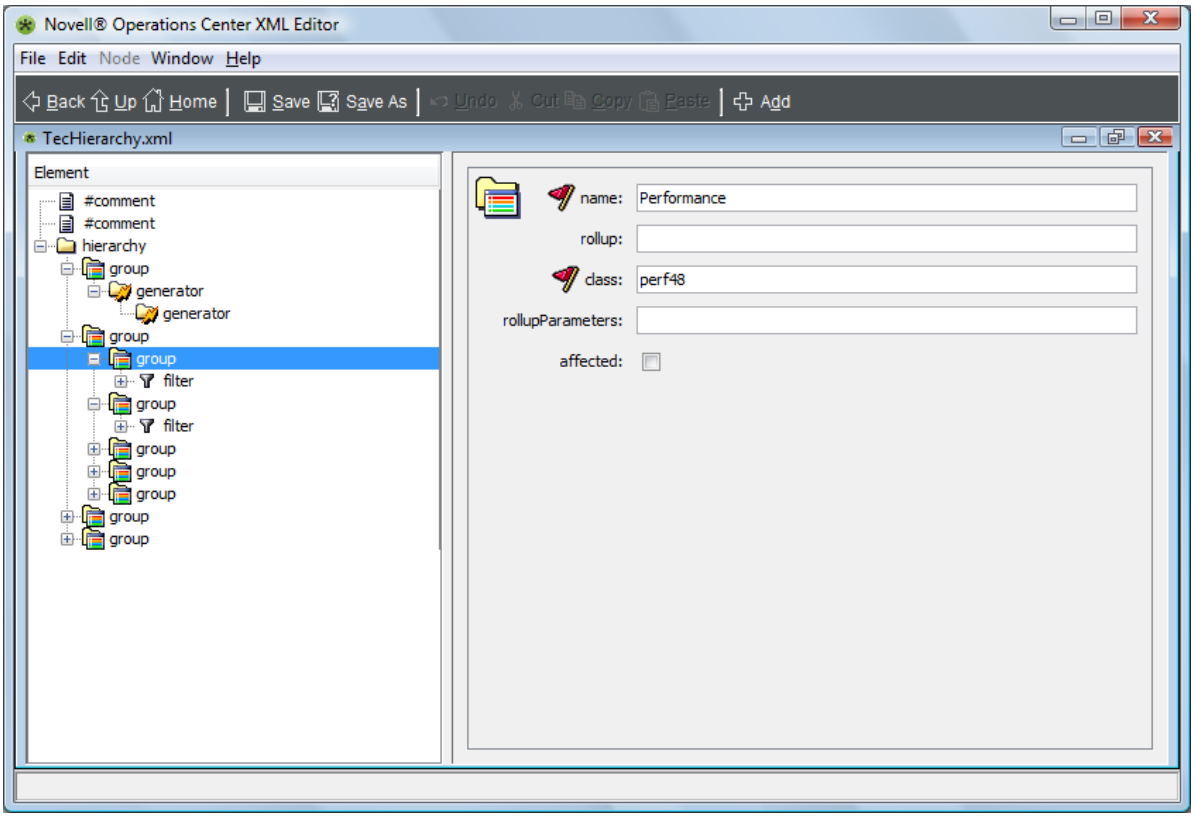
  Some management systems cannot identify the discrete element origins of the events that they generate. Moreover, these managers do not have the capacity to sort the events into a relationship structure that users can easily understand.

  The HierarchyFile is a Operations Center mechanism used by an adapter to interpret and organize the events reported by management systems that cannot identify discreet element origins. Without a HierarchyFile, Operations Center would represent as a single element everything reported by these management systems.
- Customize existing algorithm definitions or create additional algorithms.

The View Builder editor performs the same functions as the XML Editor. You can use the View Builder Repository to create, edit, and share element hierarchies in the *Service Models* hierarchy. For more information, see "Using The View Builder Repository"in the *Operations Center 5.5 Service Modeling Guide*.

The XML Editor consists of an *XML Element* hierarchy pane and an *Attribute* pane. When you select an element on the left pane, its attributes display in the right pane:

Figure A-1 *XML Editor: Select an element in the left pane and view its attributes in the right pane.*



To access the XML Editor to edit an *Adapter* hierarchy, do one of the following:

- In the *Explorer* pane, expand *Administration* > *Adapters*, right-click an adaptor that supports a hierarchy file, then select *Edit Hierarchy Definition*.

  The XML Editor loads the associated adapter hierarchy file.

  For a list of adapters that support a HierarchyFile, see the *Operations Center 5.5 Adapter and Integration Guide*.
- To customize existing algorithm definitions or create additional algorithms, see Section 15.4, “Modifying the Algorithm Library,” on page 242 for instructions on copying the original algorithms file before editing.
- It is possible to open and work with multiple XML files in the editor.

A.2 Managing XML Files

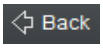
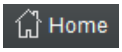
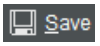
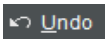
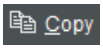
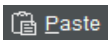
XML files can be managed through the toolbar icons or *File* menu options that the XML Editor provides, and a Document Type Definition (DTD) is required to create or add to XML files:

- Section A.2.1, “Understanding the XML Editor Toolbar,” on page 255
- Section A.2.2, “Using the Document Type Definition (DTD),” on page 255

A.2.1 Understanding the XML Editor Toolbar

Table A-1 describes the *XML Editor* toolbar and menu options:

Table A-1 *XML Editor Options*

Option	Function
	Returns focus to the element that was previously selected.
	Select the parent of the currently selected element. Continue clicking to move up the hierarchy.
	Select the top of the hierarchy.
	Saves the XML file using the current name.
	Undo any updates or modifications made on the hierarchy.
	Cuts the selected text from the current XML file.
	Copies the selected text into the current XML file.
	Pastes the cut of copied text into the current XML file.
	Adds a new element or text to the XML file.
File > *Revert*	Abandons all changes made in the current session and opens the saved version of the XML file.
File > *Save*	Saves the XML file currently loaded in the editor.
File > *Save As*	
File > *Exit*	Closes the XML Editor.

A.2.2 Using the Document Type Definition (DTD)

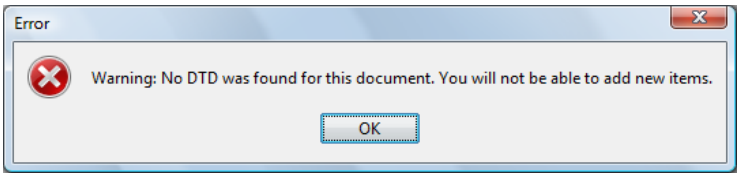
The DTD is specified at the top of an XML file to specify the valid elements for the file. The DTD is identified using the *DOCTYPE* element, as shown in the following example:

```
<!DOCTYPE hierarchy
  PUBLIC "-//Managed Object Solutions, Inc.//DTD hierarchy 2.0//EN"
"http://www.ManagedObjects.com/dtds/hierarchy_2.0.dtd">
```

All DTDs provided by Operations Center are located in the */OperationsCenter_install_path/database/examples* directory.

If an XML file that does not contain a DTD is opened in the XML Editor, the following error message displays:

Figure A-2 *Error Message: XML files with no DTD*

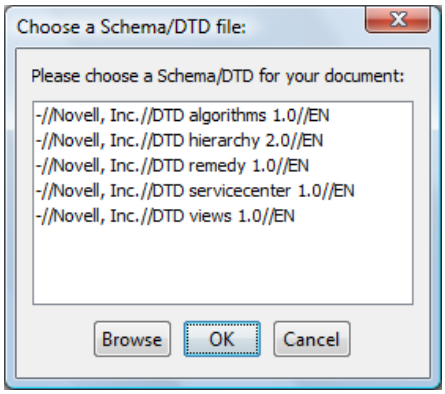


It is possible to view and edit the XML file, but not to add items to it.

To select a DTD when creating an XML file:

1 Click New on the toolbar.

The XML Editor displays the following dialog box for selecting a DTD:



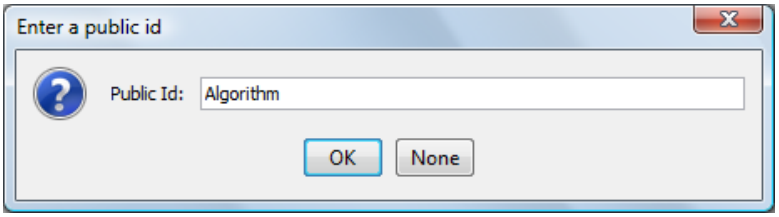
2 Do one of the following:

- Select a standard Operations Center DTD.
- Click *Browse*, select a DTD from a different location, then click *OK*.

  Operations Center provides a DTD for algorithms, hierarchy, and View Builder XML files.

  The Enter a Public ID dialog box prompts for a public ID for the DTD.
- Enter an ID.

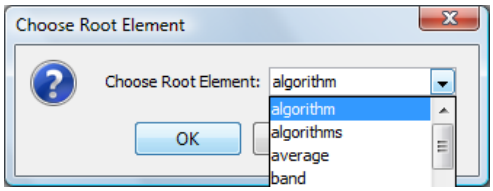
  The ID entered here displays in the Select a DTD File dialog box when the *New* icon or *File* > *New* is selected:



3 To skip creating a public ID, click *None* to open the Choose Root Element dialog box.

4 Select a root element.

The selected root element displays in the *Element* column:



5 Click *OK* to display the DTD.

A.3 Adding Elements

To add an element tag to an XML file, right-click an element in the left pane, then select *Add*. The Add submenu lists only those elements that are valid for the entry point selected. The list of elements on the *Add* menu varies, depending on the DTD associated with the XML file.

Commonly used elements tags include: `Group`, `Filter`, `Fref` (file reference), `Test`, `Properties`, and `Generator`. The *Operations Center 5.5 Adapter and Integration Guide* describes these elements in the context of the Managed Object Definition Language (MODL), which is an XML-based markup language used to create HierarchyFiles for Operations Center.

- Section A.3.1, "Adding a Tag as a Subelement," on page 257
- Section A.3.2, "Placing a New Element," on page 258
- Section A.3.3, "Adding Comments and Processing Instructions," on page 258

A.3.1 Adding a Tag as a Subelement

To add a tag as a subelement of the currently selected element:

1 Right-click an element, then select *Add*.

2 Select an item, such as *Filter*, from the submenu.

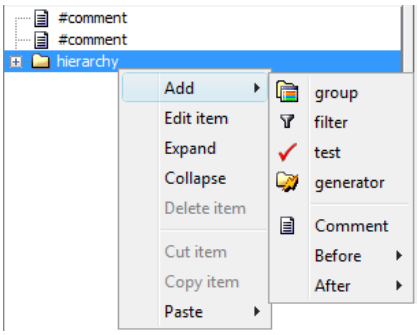
The item is added as the last subelement for the selected element.

A.3.2 Placing a New Element

To place a new element before or after the current element, at the same level as the selected element:

1. Right-click an element in the left pane, then select *Before* or *After*.
2. Click the element to add.

   It is placed above or below the current element, at the same level:



A.3.3 Adding Comments and Processing Instructions

Processing Instructions provide information to an application. The instruction begins with a target, PITarget, which identifies the application. The XML processor is required to pass these instructions to the application. For details, see the http://www.w3.org/ (http://www.w3.org/) web site.

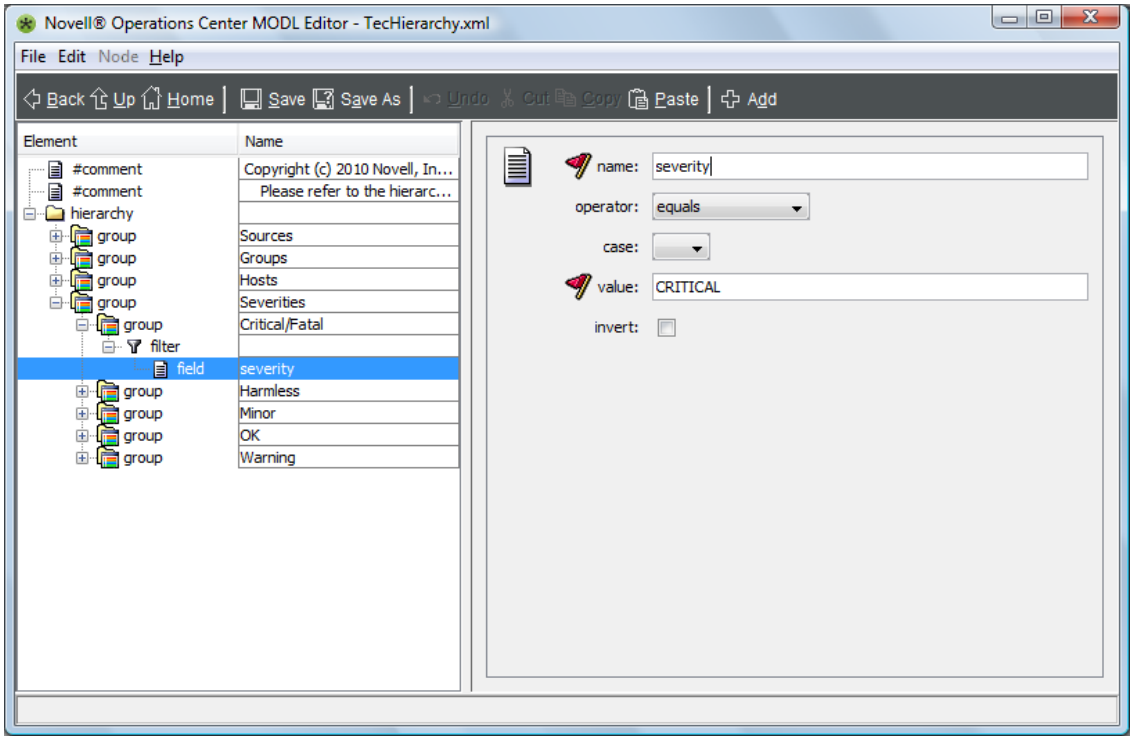
To add comments or processing instructions:

1. In the *Explorer* pane, right-click an element, then select *Add*.
2. To place a comment or processing instruction before or after the selected element, select *After* or *Before*.
3. Select *Comment or Processing Instruction*, then enter the information.

A.4 Adding and Editing Attributes

Required attributes are identified by a red flag on the *Attributes* pane. Most attribute fields allow typing values. Some have drop-down lists. Attributes with check boxes represent fields with a `Yes` or `No` value.

1 Select the check box to indicate Yes (meaning True), or leave it deselected to indicate No (meaning False):



2 When a new element is added, enter values for the required attributes.

The following message displays if you attempt to select a different element:



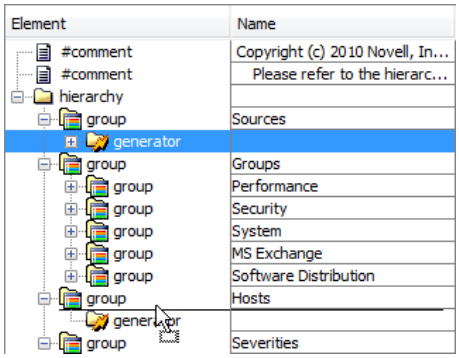
A.5 Moving Tags

XML tags can be moved using drag and drop within the current file or to the hierarchy of another XML file of the same type that is open in the editor. The rules for moving elements stem from the XML file's DTD.

- Section A.5.1, "Moving a Tag Element," on page 260
- Section A.5.2, "Cutting, Copying, and Pasting Tags," on page 260

A.5.1 Moving a Tag Element

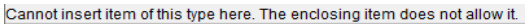
1 Drag and drop the tag to the target tag element:



A horizontal line identifies movement of an element.

The moved tag is placed as child of the target tag. A thin horizontal line visible before releasing the mouse button confirms the placement of the tag.

If an insertion is not allowed, a message is displayed in the editor's status bar:



A.5.2 Cutting, Copying, and Pasting Tags

Use the *Copy*, *Cut*, and *Paste* toolbar buttons to rearrange tag elements. *Copy* and *Paste* are also listed as options under the *Edit* menu.

Copy and *Cut* include all children of the selected tag. The *Paste* option provides the choice of pasting before or after the target tag at the same level or into the target tag as a child tag.

A.6 Deleting Tags

To delete a tag and all its children, right-click the element and then select *Delete Item*.