HP Network Node Manager i Software

For the Windows®, Linux, HP-UX, and Solaris operating systems

Software Version: 9.23

Online Help: Help for Administrators

Document Release Date: May 2013

Software Release Date: May 2013

Legal Notices

Warranty

The only warranties for HP products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HP shall not be liable for technical or editorial errors or omissions contained herein.

The information contained herein is subject to change without notice.

Restricted Rights Legend

Confidential computer software. Valid license from HP required for possession, use or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

Oracle Technology — Notice of Restricted Rights

Programs delivered subject to the DOD FAR Supplement are 'commercial computer software' and use, duplication, and disclosure of the programs, including documentation, shall be subject to the licensing restrictions set forth in the applicable Oracle license agreement. Otherwise, programs delivered subject to the Federal Acquisition Regulations are 'restricted computer software' and use, duplication, and disclosure of the programs, including documentation, shall be subject to the restrictions in FAR 52.227-19, Commercial Computer Software-Restricted Rights (June 1987). Oracle America, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

For the full Oracle license text, see the license-agreements directory on the NNMi product DVD.

Copyright Notice

Trademark Notices

Adobe® is a trademark of Adobe Systems Incorporated.

HP-UX Release 10.20 and later and HP-UX Release 11.00 and later (in both 32 and 64-bit configurations) on all HP 9000 computers are Open Group UNIX 95 branded products.

Microsoft® and Windows® are U.S. registered trademarks of Microsoft Corporation.

Oracle and Java are registered trademarks of Oracle and/or its affiliates.

UNIX® is a registered trademark of The Open Group.

Acknowledgements

This product includes software developed by the Apache Software Foundation. (http://www.apache.org)

This product includes software developed by the Indiana University Extreme! Lab. (http://www.extreme.indiana.edu)

Documentation Updates

The title page of this document contains the following identifying information:

- Software Version number, which indicates the software version.
- Document Release Date, which changes each time the document is updated.
- Software Release Date, which indicates the release date of this version of the software.

To check for recent updates or to verify that you are using the most recent edition of a document, go to:

http://h20230.www2.hp.com/selfsolve/manuals

This site requires that you register for an HP Passport and sign in. To register for an HP Passport ID, go to:

http://h20229.www2.hp.com/passport-registration.html

Or click the **New users - please register** link on the HP Passport login page.

You will also receive updated or new editions if you subscribe to the appropriate product support service. Contact your HP sales representative for details.

Support

Visit the HP Software Support Online web site at:

http://www.hp.com/go/hpsoftwaresupport

This web site provides contact information and details about the products, services, and support that HP Software offers.

HP Software online support provides customer self-solve capabilities. It provides a fast and efficient way to access interactive technical support tools needed to manage your business. As a valued support customer, you can benefit by using the support web site to:

- Search for knowledge documents of interest
- Submit and track support cases and enhancement requests
- Download software patches
- Manage support contracts
- Look up HP support contacts
- Review information about available services
- Enter into discussions with other software customers
- Research and register for software training

Most of the support areas require that you register as an HP Passport user and sign in. Many also require a support contract. To register for an HP Passport ID, go to:

http://h20229.www2.hp.com/passport-registration.html

To find more information about access levels, go to:

http://h20230.www2.hp.com/new_access_levels.jsp

Disclaimer for PDF Version of Online Help

This document is a PDF version of the online help. This PDF file is provided so you can easily print multiple topics from the help information or read the online help in PDF format.

Note: Some topics do not convert properly to PDF, causing format problems. Some elements of online help are completely removed from the PDF version. Those problem topics can be successfully printed from within the online help.

Contents

Chapter 1

Introduction for NNMi Administrators

As an NNMi administrator, you can use the console to configure the items described in the following table.

Configure NNMi

What You Can Configure	Description
Custom Polling	Using the **Custom Poller** option in the **Monitoring** folder of the **Configuration** workspace, take a proactive approach to network management by using SNMP MIB Expressions to specify additional information that NNMi should poll. You can also specify States that should be assigned to polled MIB Expression values, including any thresholds that should be set and monitored.
Custom Correlation	Using the **Custom Correlation** option in the **Incidents** folder of the **Configuration** workspace, correlate groups of incidents under a Parent Incident. This feature is useful when you want to define a relationship between a number of incidents potentially from different network objects that form a logical set to identify a problem. The set of correlations is considered complete if all of the incidents arrive within a specified time window.
Device Profiles	HP provides well over three thousand pre-configured Device Profiles, one for each known MIB-II `sysObjectID` at the time NNMi released. NNMi uses Device Profiles (which equate to `sysObjectID`) to control certain types of behavior. Using the **Device Profiles** option in the **Configuration** workspace, you can update Device Profile information. See "Configure Device Profiles" on page 292for more information.
Discovery	Using the **Discovery Configuration** option in the **Discovery** folder of the **Configuration** workspace, configure NNMi to discover only those devices that are important to you and your team. See "Discovering Your Network" on page 175 for more information. If *static* Network Address Translation (NAT), *dynamic* Network Address Translation (NAT), or *dynamic*Port Address Translation (PAT/NAPT) are used in your network management domain, see also "Overlapping Addresses in NAT Environments" on page 89.
Global Network Management	(*NNMi Advanced - Global Network Management feature*) Using the **Global Network Management** option in the **Configuration** workspace, you can configure NNMi to share the workload among multiple NNMi management servers in your network environment. See "Connecting Multiple NNMi Management Servers (NNMi Advanced)" on page 93.
ICMP and	Using the **Communication Configuration** option in the **Configuration**

Configure NNMi , continued

What You Can Configure	Description
SNMP Communication Protocols	workspace, provide the SNMPv1 or SNMPv2c community strings (read and write) for your network environment, or provide the SNMPv3 User Names for your network environment. Configure NNMi settings for timeout, retry, and port usage for ICMP and SNMP traffic. See "Configuring Communication Protocol" on page 119 for more information.
Incidents	Using the **Incidents** folder in the **Configuration** workspace, review the many predefined incident configurations provided by NNMi . Edit any of the configurations provided by NNMi or create your own . See "Configuring Incidents" on page 589 for more information.
Interface Groups	Using the **Interface Groups** option in the **Object Groups** folder of the **Configuration** workspace, identify important devices. Interface Groups are filters for interface and IP address views. Interface Groups can also control how NNMi monitors network devices. See "Create Interface Groups" on page 321 for more information.
Interface Types	Interface Type definitions cover all known industry-standard IANA ifType-MIB variables at the time of the release of NNMi. Using the **ifTypes** view in the **Configuration** workspace, add a additional `ifType` values to the NNMi list. This option is useful if your team acquires new devices that are configured with new industry-standard `ifType` values not yet preconfgiured by NNMi. See "Add New ifType Values (Interface Types) to the List" on page 333 for more information.
Management Stations (6.x/7.x)	Using the **Management Stations (6.x/7.x)** option in the **Configuration** workspace, configure how events that are received from NNM 6.x or 7.x management stations are handled by NNMi . See "Configure Remote NNM 6.x and 7.x Management Stations" on page 1221 for more information.
MIBs	Using the **MIB Expressions** option in the **MIBS** folder of the **Configuration** workspace, take a proactive approach to network management by using SNMP MIB Expressions to specify additional information that NNMi should poll. See "Configure MIB Expressions" on page 1473 for more information. Using the **MIBs** folder, you can also view and configure the following: • Loaded MIBs • MIB Variables • MIB Notifications • MIB Textual Conventions • MIB OID Types
Monitoring	Using the **Monitoring Configuration** option in the **Monitoring** folder of the **Configuration** workspace, define how and how often important devices are monitored by NNMi . See "Monitoring Network Health" on page 340 for more information.

Configure NNMi , continued

What You Can Configure	Description
Node Groups	Using the **Node Groups** option in the **Object Groups** folder of the **Configuration** workspace, identify important devices. You can then filter node, interface, IP address, and incident views by Node Group. You can also specify Node Groups when configuring monitoring and incidents. See "Create Node Groups" on page 295 for more information.
Node Group Map Settings	Using the **User Interface Configuration** option in the **Configuration** workspace, specify the Node Group map configuration including the Node Group and background image to be used in a Node Group map. See "Define Node Group Map Settings" on page 488 for more information.
Object Groups	Using the **Node Groups** and **Interface Groups** options in the **Object Groups** folder of the **Configuration** workspaces, define groups of nodes or interfaces. Use these object groups as filters to quickly locate information in views. See "Creating Groups of Nodes or Interfaces" on page 294 for more information. You can also monitor the health of each group, see "Configure NNMi Monitoring Behavior" on page 340.
Route Analytic Management Servers (RAMS)	(*NNMi Advanced*) Using the **RAMS Servers** option in the **Configuration** workspace, configure sources of Route Analytics Management Systems data for NNMi to use. See "Using Route Analytics Management Systems (RAMS) with NNMi Advanced" on page 1407.
Security	Using the **Security** option in the **Configuration** workspace, control access to NNMi. See "Configuring Security" on page 503 for more information. **Tip:** NNMi can be configured to use the Lightweight Directory Access Protocol (LDAP) and X.509 Certificates such as Public Key Infrastructure (PKI) user authentication for NNMi user names, passwords, and User Group Membership assignments. Additional steps are required. See "Choose a Mode for NNMi Access" on page 504.
Status	Using the **Status Configuration** option in the **Configuration** workspace, configure how Node Group Status is calculated. You can choose to assign the Node Group the most severe status of any Node Group member or configure the percentage thresholds for one or more Node Group target statuses. See "Configure Node Group Status" on page 318for more information.
Trap Forwarding	Using the **Trap Forwarding Configuration** option in the **Trap Server** folder under the **Incidents** folder of the **Configuration** workspace, configure trap forwarding filters and destinations. See "Configure Trap Forwarding" on page 1376 for more information.
Trap Logging	Using the **Trap Logging Configuration** option in the **Trap Server** folder under the **Incidents** folder of the **Configuration** workspace, configure how you want trap information to appear in the trap logging file. See "Configure Trap Logging" on page 1387for more information.

Configure NNMi , continued

What You Can Configure	Description
User Interface	Using the **User Interface Configuration** option in the **Configuration** workspace, configure the following user interface features: - User accounts - Default map settings - Node Group map settings - Default Line Graph settings - Menus and menu items - Icons displayed for Device Profiles

NNMi provides a variety of tools to assist you with these configuration tasks. Each of these tools is described in the following table. You can extend NNMi using HP Network Node Manager i Software Smart Plug-ins (iSPIs) as described in "Extending NNMi Capabilities" on page 1414.

NNMi Administrator Tools

Tool	Description
Actions	Used to perform automated tasks on a single object or on a group of objects. For example, you can use the Actions menu to change the Management Mode of one or more nodes from **Managed** to **Out of Service**. Actions are available from table views, map views, and forms. See "Actions Provided by NNMi" on page 43 for more information
Configuration Workspaces	The console provides a workspace for each kind of item you can configure in NNMi . See the preceding "Configure NNMi " table for more information.
Lookup Fields	Provided in forms, fields that include the ▾ icon provide access to a list of all available attribute values, and in some locations enable you to create attribute values. See "Lookup Fields" on page 40 for more information.
NNMi Processes and Services	NNMi is built on a group of processes and services. You can list these processes and services. You can stop and start individual processes and services. See "NNMi Processes and Services" on page 81 for more information.
Tools	Used to access the following types of information: - NNMi status and monitoring information - Trap analysis information - User information and log files - Security configuration reports - MIB Browser - Attached switch port information for a selected Node

Administrator Tools in the Console

When configuring settings for NNMi, you create configuration object instances. For example, to create a new URL action, you must create a new URL action instance. As another example, to specify configuration settings for discovery, you might create object instances that contain ranges of IP addresses that you want NNMi to use as hints for Spiral Discovery.

The console provides the following tools to assist you with configuration tasks:

- "Configuration Workspaces" on the next page
- "Lookup Fields" on page 40
- "Create a Configuration Object Instance Using the Form Toolbar" on page 42
- "Delete One or More Objects" on page 1604

Quick Start Configuration Wizard

Before you use the Quick Start Configuration Wizard, review “Using the Quick Start Configuration Wizard” in the *NNMi Interactive Installation Guide*. To access the *NNMi Interactive Installation Guide*, follow these steps:

1. Unzip the `nnmi_interactive_installation_en.zip` file located in the top level directory of the NNMi 9.20 installation media.
2. Double-click `nnmi_interactive_installation_en.htm`.

The Quick Start Configuration Wizard automatically runs immediately after Network Node Manager (NNMi) installation completes. Use the Quick Start Configuration Wizard to configure NNMi in a limited (or test) environment. The Quick Start Configuration Wizard helps you to complete the following initial set up tasks:

- Provide the *read community strings* for your SNMPv1 or SNMPv2c environment to enable "Get" commands
- Provide the USM settings for your SNMPv3 environment
- Discover a limited range of network nodes
- Set up an initial administrator account

You can launch the wizard using the following URL:

`http://<serverName>:<portNumber>/quickstart/`

> **Note:** If the NNMi Web server uses the https protocol, use `https` instead of `http`. See the "Working with Certificates for NNMi" chapter in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.

`<serverName>` = the fully-qualified domain name of the NNMi management server (values allowed here are determined by the *Enable URL Redirect* setting in User Interface Configuration, see "Configuring the NNMi User Interface" on page 467)

`<portNumber>` = the NNMi HTTP port number

> **Note:** HP recommends that you run the Quick Start Configuration Wizard only one time immediately after NNMi installation.

After using the Quick Start Configuration Wizard to set up a test network, see "Configuration Workspaces" below for information about completing additional NNMi configuration tasks.

Configuration Workspaces

NNMi administrators use the Configuration workspaces to configure the following items related to NNMi.

> **Note:** On tables in configuration forms, if the cursor changes to indicate a hyperlink when you mouse over a column heading, you are able to sort the column's data. You cannot change the sort on some of the tables on the forms in the configuration workspace.

NNMi Configuration Workspaces

Name	Description
Communication Configuration	Use to configure how NNMi uses ICMP and SNMP in your network environment. See "Configuring Communication Protocol" on page 119.
Discovery → Discovery Configuration	Use to specify the devices to be discovered. See "Discovering Your Network" on page 175.
Discovery → Seeds	A discovery seed is a specific node that you want NNMi to discover. Discovery seeds are sometimes optional and sometimes required. See "Specify Discovery Seeds" on page 256.
Discovery → Tenants	Each Node must be assigned to a Tenant. NNMi provides a Tenant named Default Tenant. NNMi administrators can create additional Tenant objects as needed. *Auto-Discovery* is available only for the Default Tenant. See "Configure Tenants" on page 194. **Note:** If your network management environment includes overlapping address domains, you must configure each domain as a unique Tenant.
Discovery → Overlapping Address Mappings	If *static* Network Address Translation (NAT) is part of your network management domain, and the NNMi management server is outside of that static NAT domain, you can configure NNMi to display the NAT *external IP address* (public address) in the Mapped Address attribute of the IP Address form for a NAT *internal IP address* (such as a private IPv4 address) pair. See "Overlapping Address Mapping" on page 191.
Monitoring → Monitoring Configuration	Use to enable the NNMi State Poller. See "Monitoring Network Health" on page 340.

NNMi Configuration Workspaces, continued

Name	Description
Monitoring → Custom Poller Configuration	Use to configure SNMP MIB Expressions that specify additional information NNMi should poll. See "Create Custom Polling Configurations" on page 419
Incidents → Incident Configuration	Use to specify the information displayed with an incident, including its name, the message you want to be displayed, the way it should be categorized, its initial status, and how you want to identify duplicate traps. See "Configuring Incidents" on page 589.
Incidents → SNMP Trap Configurations	Use to configure incidents that originate from an SNMP trap.
Incidents → Syslog Message Configurations	HP ArcSight. Use to map syslog information to a Syslog Message incident configuration.
Incidents → Management Event Configurations	Use to configure incidents that are generated from the NNMi Causal Engine.
Incidents → Pairwise Configurations	Use the Pairwise Configuration to pair the occurrence of one incident with another subsequent incident. See "About Pairwise Configurations" on page 660.
Incidents → Custom Correlation Configuration	Use to correlate groups of incidents under a Parent Incident.
Incidents → Trap Server → Trap Forwarding Configuration	Use to forward SNMP trap to other servers in your network environment. See "Configure Trap Forwarding" on page 1376.
Incidents → Trap Server → Trap Logging Configuration	Use to configure how SNMP traps should appear in the `trap.log` and `trap.csv` log files. See "Trap Logging Configuration Form" on page 1388
Status Configuration	Use to configure Node Group status calculations using either of the following methods: • Assign the Node Group the most severe status of any Node Group member. This is the default. • Configure the percentage thresholds for one or more Node Group target statuses. See "Configure Node Group Status" on page 318.

NNMi Configuration Workspaces, continued

Name	Description
Global Network Management	(*NNMi Advanced - Global Network Management feature*) Use to configure communication between Global Managers and Regional Managers in your network environment. See "Connecting Multiple NNMi Management Servers (NNMi Advanced)" on page 93.
User Interface → User Interface Configuration	Use to configure many user interface features: • The NNMi console timeout interval. • The initial view that you want NNMi to display. • Specify that NNMi users must provide one of the following in the URL for accessing NNMi: ▪ The Fully Qualified Domain Name (FQDN) of the NNMi management server. ▪ Any hostname or IP address associated with the NNMi management server (NNMi automatically redirects these to the FQDN) • Whether NNMi displays unlicensed features that require a special license, such as NNMi Advanced. See "Configuring the NNMi User Interface" on page 467. **Default Map Settings** tab - Use to configure the default settings for map views. These settings can be overridden for a specific map using the Node Group Map Settings tab. See "Configure Maps" on page 487. **Default Line Graph Settings** tab - Use to configure the SNMP MIB data that you want to make available to your network operators in a graph format. This graph is available through the Actions menu and displays in real time. See "Configure Default Settings for Line Graph" on page 472. **Tip:** You can right-click any object in a table or map view to access the **Actions** menu.
User Interface → Node Group Map Settings	- Use to specify the Node Group and background image to be used in a Node Group map. Map settings include the following: • Node group name • The order in which Node Group maps should appear in the Topology workspace • Minimum User Group for saving edited locations for each node in the map • Refresh information • Connectivity information • Background image URL • Background image scale

NNMi Configuration Workspaces, continued

Name	Description
User Interface → Menus	Use to configure how menu items are nested in the NNMi console. See "Configure Menus" on page 502.
User Interface → Menu Items	Use to make changes or additions to the items available in the Actions menu. See "Configure Menu Items" on page 502 for more information.
User Interface → Icons	Use to customize the icons associated with a Device Profile or specific Nodes. These icons appear in table views, menu items, and as foreground images on an NNMi topology map. See "Customize Device Profile Icons" on page 474.
Security	Use to map the following objects to control access to the network: • Users to User Groups • User Groups to Security Groups • Security Groups to Nodes See "Configuring Security" on page 503
MIBs → Loaded MIBs	Use to determine the MIBs loaded on the NNMi management server. See "Examine Available MIBs and MIB Variables" on page 1450.
MIBs → MIB Variables	Use to determine the MIB Variables supported for a selected node. See "Determine the MIB Variables Supported for a Node (for Administrators)" on page 1454
MIBs → MIB Notifications	Enables you to view the SNMP trap information, if any, that is defined by the selected MIB. See "MIB Notification Form (for Adminstrators)" on page 1467
MIB → MIB Textual Conventions	Use to examine the format rules for the selected Textual Convention that are defined in the MIB. NNMi uses these MIB format rules to determine how to display any associated MIB variable values of type Octet String. See "MIB Textual Conventions Form" on page 1470.
MIBs → MB Expressions	Use to determine the MIB Expressions available for Custom Poller or Line Graphs. See "Create a Custom Poller Collection" on page 421 and "Configure SNMP Line Graph Actions" on page 1437.
MIBs → MIB OID Types	If you find that the results of a MIB Expression displayed in a Line Graph or a Gauge or used by Custom Poller are not as expected, use the MIB OID Types configuration to override values for the following items for a MIB Object Identifier (OID)
MIBs → ifTypes	Use to determine the list of available interface types. NNMi administrators use these ifType values to define Interface Groups. See "Add New ifType Values (Interface Types) to the List" on page 333.
Device Profiles	Use to see and edit device profile information. Device profile information includes the SNMP object ID, model, and vendor. See "Configure Device Profiles" on page 292.

NNMi Configuration Workspaces, continued

Name	Description
Object Groups → Node Groups	Use to group your devices for viewing and monitoring purposes. See "Create Node Groups" on page 295.
Object Groups → Interface Groups	Use to group your devices for viewing and monitoring purposes. See "Create Interface Groups" on page 321.
RAMS Servers	(*NNMi Advanced*) Use to configure sources of Route Analytics Management Systems data for NNMi to use. See "Using Route Analytics Management Systems (RAMS) with NNMi Advanced" on page 1407.

Enable or Disable Configurations

Using the **Actions** menu, you can enable or disable one or more of the following configurations:

Note: When you enable or disable a configuration, NNMi assigns the value **Customer** as the Author name. See Author form for important information.

Enable or Disable NNMi Configurations

Configuration	Configuration Workspace Option
SNMP Trap Incidents	Incidents
Syslog Messages Incidents	Incidents
Remote NNM 6.x/7.x Event Incidents	Incidents
Management Event Incidents	Incidents
Pairwise	Pairwise Configuration
Menus	User Interface Configuration
Menu Items	User Interface Configuration

To enable an NNMi configuration:

1. Navigate to the table view of the configurations you want to change. For example, select **User Interface Configuration** from the **Configuration** workspace and select the **Menus** tab.
2. To enable a configuration, select the row representing the configuration you want to enable.
3. Select **Actions** → **Enable Configuration**.

   If you are in the configuration form, NNMi selects Enabled ☑.

   If you are in the table view, NNMi displays a ✓ check in the Enabled column for each instance selected.

To disable an NNMi configuration:

1. Navigate to the table view of the configurations you want to change. For example, select **User Interface Configuration** from the **Configuration** workspace and select the **Menus** tab.

2. Do one of the following:

   a. To disable a configuration, select the row representing the configuration you want to edit.

   b. To disable more than one configuration, press CTRL-Click and select each row that represents a configuration instance that you want to disable.

3. Select **Actions** → **Disable Configuration**.

   If you are in the configuration form, NNMi removes the check mark from Enabled □.

   If you are in the table view, NNMi removes the ✓ check mark in the Enabled column for each instance selected.

Lookup Fields

Lookup fields have the following icon: .

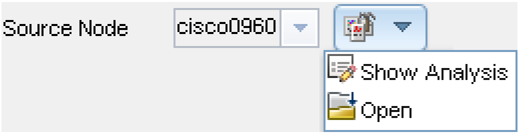
The Lookup field represents an associated object instance. For example, an Incident form has an associated Source Node attribute. Information about this source node is available in and accessed through the Lookup field.

Possible Drop-Down Menu Options in Lookup Fields

Option	Description
Show Analysis	Display Analysis Pane information for the selected object. (See Use the Analysis Pane for more information about the Analysis Pane.)
Quick Find	Display a list of valid choices for populating the current attribute field.
Open	Open the form for the related object instance that is currently selected in the lookup field. Review all attributes of the related object. Depending on your role, you can edit these attributes.
New	Create a new object instance to relate to the current object.

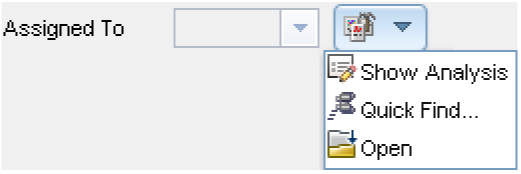
You can use Lookup fields in a variety of ways:

- **Read-only fields - to provide additional information about the associated object**. Click Show Analysis (Use the Analysis Pane) or Open to see the details of this object.

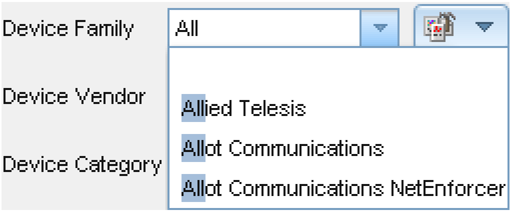
  

  Source Node cisco0960

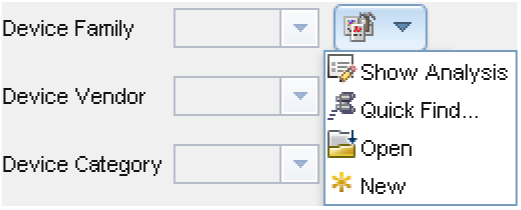
- **Selection fields - to change the association to another object instance**. Click Quick Find to select from a list of previously configured objects ("Use the Quick Find Window" on the next page).



Or type a case-sensitive string into the input box ("Use Autocomplete" on the next page).



- **Read-write fields - create an entirely new object instance for this association**. Click New. An empty form opens for you to fill in, creating a new object instance.

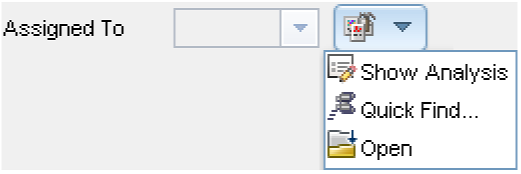


Use the Quick Find Window

The Quick Find option is available only in Lookup fields that are modifiable. Use the Quick Find option to see the list of available object instances appropriate for populating the current Lookup field.

To list all existing object instances that could be related to the current object:

1. From the lookup field of interest, click the Look up icon:

2. Select Quick Find.

   NNMi displays a table view of object instances that are available to associate with to the current object instance.

3. In the Quick Find window, do one of the following:

Clear	Click the **Clear** button to remove an association with this object. The Quick Find window closes, and the current lookup field is empty.
OK	Select a row in the table, and click the **OK** button. The Quick Find window closes, and the object instance you selected populates the current lookup field.

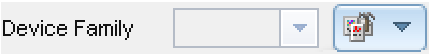
Cancel	Click the **Cancel** button to return to the previous form without making any changes

Use Autocomplete

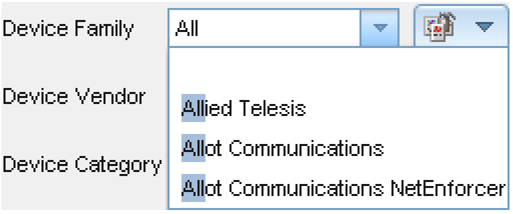
The autocomplete feature is available only in Lookup fields that are modifiable. As you type, NNMi lists the available object instances for populating the current Lookup field.

To use the autocomplete feature:

1. Start typing the first few letters (case-sensitive) of the name of the object you want to associate with the current one.

   The Lookup field displays a drop-down list below the input field. This list includes all potential existing objects with names that match the letters as you enter them.

2. Use the scroll arrows or the mouse to select from the displayed list.

   The selected object populates the Lookup field and is now associated with the current object.

Create a Configuration Object Instance Using the Form Toolbar

You can save time by generating a new form from within another form. The new form is based on the object type for the original form and contains only the default values set by NNMi for particular attributes for that object. Any attributes that have no default value appear blank.

This tool is useful when you want to create multiple object instances that have similar attribute values.

To create a new object instance using the form toolbar:

1. Open the form representing the object of interest.
2. From the form toolbar, click the Save and New icon.

   A new form appears that contains the default attribute values for the object type represented by the original form.

3. Select the **Save and Close** icon to save your changes and return to the view.

Delete One or More Objects

Each row in a table view and each symbol in a map view represents an instance of the object type being displayed. For example, in a node view, each row of the table represents an instance of a node in your network.

Some NNMi users can delete object instances. For example, you might need to delete a node that is no longer being managed. See "Delete Nodes" on page 1602 for more information.

To delete an object instance:

1. Select the object of interest:
   - In a table view, select the row that represents the object.
   - In a map view, click the map symbol.
   - In a form, proceed to step 2.
2. To delete the object, click the ✖ Delete icon.

   The object is deleted from the NNMi database and removed from the current view.

To delete multiple object instances:

1. Select the objects of interest:
   - In a table view, press CTRL-Click and select each row that represents an object you want to delete.
   - In a map view, CTRL-Click each map symbol.
2. To delete the objects, click the ✖ Delete icon.

   > **Note:** For Node objects, you can use this method to delete up to 20 nodes at one time. To delete more than 20 nodes, see the nnmnodedelete.ovpl Reference Page.

   > **Tip:** For all other objects, you can delete any number.

   Each object is deleted from the NNMi database and removed from the current view.

Related Topics

Using Table Views

Using Map Views

"Configure Whether to Delete Unresponsive Nodes" on page 212

Actions Provided by NNMi

Note: (*NNMi Advanced - Global Network Management feature*) If your NNMi console is a Global Manager and the selected node is being managed by a Regional Manager (another NNMi management server in your network environment), some actions are not available.

The following tables describe the actions provided by NNMi:

As shown in the table, the actions available depend on the object selected.

> **Tip:** You can right-click any object in a table or map view to access the **Actions** menu.

Note the following:

- The Default NNMi Role determines the Actions displayed.
- The Minimum NNMi Role determines the lowest NNMi Role to which the Action can be configured.
- The Default Object Access Privileges determines the Actions a user can execute.
- As the NNMi Adminsitrator, you determine a user's NNMi Role and Object Access Privileges. See "Configuring Security" on page 503for more information.

Actions Provided for Incidents

Action	Description	NNMi Role Default/Minimum	Default Object Access Privilege
Node Actions	Provides access to all of the actions available for a the Incident's Source Node. See Actions Provided for Nodes for more information.	See Actions Provided for Nodes.	See Actions Provided for Nodes.
Interface Actions	**Only available for incidents with the Source Object attribute value set to Interface.** Provides access to all of the	See Actions Provided for Interfaces.	See Actions Provided for Interfaces

Actions Provided for Incidents , continued

Action	Description	NNMi Role Default/Minimum	Default Object Access Privilege
	actions available for an interface. See Actions Provided for Interfaces for more information.		
IP Address Actions	**Only available for incidents with the Source Object attribute value set to IP Address.** Provides access to all of the actions available for an IP address. See Actions Provided for Addresses for more information.	See Actions Provided for IP Addresses.	See Actions Provided for IP Addresses
Node Group Map	**Maps → Node Group Map** Displays the lowest level Node Group map to which the Source Node belongs. For example, if the node belongs to a *Child* Node Group, the *Child* Node Group displays. See Node Group Maps. **Note**: If the Source Node is a member of more than one Node Group, NNMi displays the list of possible Node Groups. Right-click the Node Group of interest and select **Maps** > **Node Group Map**. If the incident's Source Object is an Island Node Group, NNMi displays the Island Node Group map. See "Island Node Groups" on page 337. **Note**: Incidents with the Source Object attribute value set to Island Node Group include **Remote site** in the incident message. See Island Node Group Map for more information. When the selected Source Node is not a member of any Node Group, and you select the **Node Group**	Operator Level 1/ Operator Level 1	Object Operator Level 1

Actions Provided for Incidents , continued

Action	Description	NNMi Role Default/Minimum	Default Object Access Privilege
	Map action, NNMi displays an information message.		
Path View	**Maps → Path View** Displays a map showing the route between two specified nodes, using the Source Node as the starting point. **Note**: *NNMi Advanced*. Path View works only with IPv4 addresses. The NNMi Advanced IPv6 address values are not valid choices for Path View. Any devices in your network that are configured with IPv6 addresses cannot be displayed on Path View maps.	Operator Level 1/ Guest	Object Operator Level 1
Source Node	**Source Node** Displays the Node form of the Source Node object instance.	Operator Level 1/ Operator Level 1	Object Operator Level 1
Source Object	Displays the form of the source object instance.	Operator Level 1/ Operator Level 1	Object Operator Level 1
Node Group Members	**Node Group Members** *Island Node Group incidents only*. Displays a table of the nodes that are members of the Island Node Group that is the Source Object for the selected incident. See "Island Node Groups" on page 337. **Note**: Incidents with the Source Object attribute value set to Island Node Group include **Remote site** in the incident message.	Operator Level 1/ Operator Level 1	Object Operator Level 1
Graph	**Graph Custom Poller Results**	Operator Level 1/ Operator Level 1	Object Operator

Actions Provided for Incidents , continued

Action	Description	NNMi Role Default/Minimum	Default Object Access Privilege
Custom Poller Results	Graphs all MIB expressions from each of the Custom Poller Collections associated with the selected incident's Source Node.		Level 1
Ping	**Node Access** > **Ping** Tests whether a node or IP address is reachable using the ping command from the NNMi console.	Operator Level 1/Operator Level 1	Operator Level 1
Open Web Page	**Node Access** > **Open Web Page** Opens the default Web page for the selected node.	Administrator/Administrator	
Trace Route	**Node Access** > **Trace Route** Trace the route path to identify bottlenecks along the destination path provided.	Operator Level 1/Operator Level 1	Operator Level 1
Telnet	**Node Access** > **Telnet** Establish a connection to a node to view or change configuration information	Operator Level 2/Operator Level 2	Operator Level 2
Secure Shell	**Node Access** > **Secure Shell** Establish a connection to a node to view or change configuration information.	Operator Level 2/Operator Level 2	Operator Level 2
Delete	**Delete** Deletes the selected Incident object or objects (maximum 20). To delete more than 20 nodes, see the nnmnodedelete.ovpl Reference Page.	Administrator/Administrator	Object Administrator
In Progress	**Change Lifecycle** → **In Progress** Changes the lifecycle state to **In Progress** for the selected incident.	Operator Level 1/Operator Level 1	Object Operator Level 1
Completed	**Change Lifecycle** → **Completed** Changes the lifecycle state to **Completed** for the selected	Operator Level 1/Operator Level 1	Object Operator Level 1

Actions Provided for Incidents , continued

Action	Description	NNMi Role Default/Minimum	Default Object Access Privilege
	incident.		
Close	**Change Lifecycle → Close** Changes the lifecycle state to **Closed** for the selected incident.	Operator Level 1/Operator Level 1	Object Operator Level 1
Assign Incident	**Assign → Assign Incident** Displays a list of registered users to select from. This user name appears in the **Assigned To** column of the incident view.	Operator Level 1/Operator Level 1	Object Operator Level 1
Own Incident	**Assign → Own Incident** Assigns the incident to the current user. This user name appears in the **Assigned To** column of the incident view.	Operator Level 1/Operator Level 1	Object Operator Level 1
Unassign Incident	**Assign → Unassign Incident** Removes the user name from the **Assigned To** column of the incident view.	Operator Level 1/Operator Level 1	Object Operator Level 1
Incident Configuration Reports	Displays a report of the configuration settings that define this Incident. See "View an Incident Configuration Report" on page 1372 for more information.	Administrator/Administrator	Object Administrator
Open Incident Configuration	Displays the selected Incident's configuration form.	Administrator/Administrator	Object Administrator
Run Diagnostics (iSPI NET only)	(*HP Network Node Manager iSPI Network Engineering Toolset Software*) When installed, NNM iSPI NET gathers diagnostic information from the Source Node.	Operator Level 1/Operator Level 1	Object Operator Level 1

Actions Provided for Trap Logging Configuration

Action	Description	NNMi Role Default/Minimum	Default Object Access Privilege
Show SNMP Trap Configuration	Displays the SNMP Trap Incident Configuration form, if any, for the current Trap Logging Configuration. The Configuration form displayed is for the SNMP Trap Incident associated with the Trap Logging Configuration.	Administrator/Administrator	Administrator

Actions Provided for Nodes

Action	Description	NNMi Role Default/Minimum	Default Object Access Privilege
Layer 2 Neighbor View	**Maps** > **Layer 2 Neighbor View** Represents your network's physical connections and LAN switch traffic routes.	Operator Level 1/ Guest	Object Operator Level 1
Layer 3 Neighbor View	**Maps** > **Layer 3 Neighbor View** Represents your network's router traffic.	Operator Level 1/ Guest	Object Operator Level 1
Node Group Map	**Maps** > **Node Group Map** Displays the lowest level Node Group map to which the selected Node belongs. For example, if the node belongs to a *Child* Node Group, the *Child* Node Group displays. See Node Group Maps. If the Node is a member of more than one Node Group, NNMi displays the list of possible Node Groups. Right-click the Node Group of interest and select **Maps** > **Node Group Map**. When the selected Source Node is not a member of any Node Group, and you select the **Node Group Map** action, NNMi displays an information message.	Operator Level 1/ Operator Level 1	Object Operator Level 1
Path View	**Maps** > **Path View** Displays a map showing the route between two specified nodes, using the Source Node as the starting point.	Operator Level 1/ Guest	Object Operator Level 1

Actions Provided for Nodes , continued

Action	Description	NNMi Role Default/Minimum	Default Object Access Privilege
	Note: *NNMi Advanced*. Path View works only with IPv4 addresses. The NNMi Advanced IPv6 address values are not valid choices for Path View. Any devices in your network that are configured with IPv6 addresses cannot be displayed on Path View maps.		
Graphs	Displays a pre-configured graph of real-time data for a selected node. NNMi provides a set of Line Graph that are configured to display real-time SNMP data. See Line Graphs Provided by NNMi for more information. Line Graph graphs can also come from the following sources: • Your NNMi administrator might configure additional graphs. • NNM iSPI software.	Operator Level 1/ Guest	Object Operator Level 1
Ping (from server)	**Node Access → Ping (from server)** Tests whether a node is reachable using the ping command. (*NNMi Advanced*) If the Global Network Management feature is enabled and you are signed into a Global Manager: • Node managed by the Global Manager = Actions → Ping issues an ICMP request from the Global Manager (NNMi management server). • Node managed by a Regional Manager = Actions → Ping accesses that Regional Manager (NNMi management server) and issues the ICMP request. **Note:** You must sign into that Regional Manager unless your network environment enables Single Sign-On (SSO) to that Regional Manager through the Global Manager. For more information, see the “Configuring Single Sign-On for Global	Operator Level 1/ Operator Level 1	Object Operator Level 1

Actions Provided for Nodes , continued

Action	Description	NNMi Role Default/Minimum	Default Object Access Privilege
	Network Management" section in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.		
Open Web Page	**Node Access** > **Open Web Page** Opens the default Web page for the selected node.	Administrator/ Administrator	
Trace Route (from server)	**Node Access** → **Trace Route (from server)** Traces a route path from the using the traceroute command. (*NNMi Advanced*) If the Global Network Management feature is enabled and you are signed into a Global Manager: • Node managed by the Global Manager = **Actions** → **Node Access** → **Trace Route** issues a request from the Global Manager (NNMi management server). • Node managed by a Regional Manager = **Actions** → **Node Access** → **Trace Route** accesses that Regional Manager (NNMi management server) and issues the request in a manner appropriate for the operating system in use on the Regional Manager. **Note:** You must sign into that Regional Manager unless your network environment enables Single Sign-On (SSO) to that Regional Manager through the Global Manager. For more information, see the "Configuring Single Sign-On for Global Network Management" section in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.	Operator Level 1/ Operator Level 1	Object Operator Level 1

Actions Provided for Nodes , continued

Action	Description	NNMi Role Default/Minimum	Default Object Access Privilege
Telnet (from client)	**Node Access → Telnet (from client)** Uses Transmission Control Protocol (TCP) protocol from the computer that launched your current browser (not the NNMi management server) to open a Telnet (teletype network) virtual terminal command-line interface from the selected node or Source Node of the selected object. See Establish Contact with a Node.	Operator Level 2/ Operator Level 2	Object Operator Level 2
Secure Shell (from client)	**Node Access → Secure Shell (from client)** Uses Secure Shell (SSH) protocol from the computer that launched your current browser (not the NNMi management server) to open a Secure Shell virtual terminal command-line interface from the selected node or Source Node of the selected object. See Establish Contact with a Node.	Operator Level 2/ Operator Level 2	Object Operator Level 2
Status Poll	**Polling → Status Poll** Instructs NNMi to gather real-time data for all the information that NNMi uses to calculate Status for each selected Node (maximum 10). A window for each Node displays with a report about which information was gathered. The NNMi administrator determines the list of information gathered by establishing Monitoring configuration settings. See "Monitoring Network Health" on page 340 for more information. Note the following: • Status Poll might cause an object's Status to be updated.To see the resulting Node status, see Verify Current Status of a Device. • Using **Actions → Status Poll** does not affect the timing of the Polling interval configured for the device. **Tip**: The nnmstatuspoll.ovpl command line tool does the same thing as **Actions → Status Poll**. (*NNMi Advanced*) If the Global Network Management feature is enabled and you are signed into a Global Manager:	Operator Level 2/ Operator Level 2	Object Operator Level 2

Actions Provided for Nodes , continued

Action	Description	NNMi Role Default/Minimum	Default Object Access Privilege
	• Node managed by the Global Manager = Actions → Status Poll requests that the Global Manager (NNMi management server) perform a status poll on the node. • Node managed by a Regional Manager = Actions → Status Poll requests that the Regional Manager perform a status poll on the node, the Global Manager displays the results. Latest Status Poll results are available on both NNMi management servers (Global and Regional). **Note:** You do not need to sign-in to the Regional Manager.		
Configuration Poll	**Polling → Configuration Poll** Runs a real-time configuration check of the selected device to detect any changes since the last discovery cycle. (*NNMi Advanced*) If the Global Network Management feature is enabled and you are signed into a Global Manager: • Node managed by the Global Manager = **Actions → Polling → Configuration Poll** results are provided by the Global Manager (NNMi management server). • Node managed by a Regional Manager = **Actions → Polling → Configuration Poll** requests an updated *copy* of the configuration information from the Regional Manager, then the Global Manager displays the results. **Note:** You do not need to sign-in to the Regional Manager.	Operator Level 2/ Operator Level 2	Object Operator Level 2
Communication Settings	**Configuration Details → Communication Settings** Displays the communication configuration information for the selected node.	Administrator/ Administrator	Object Administrator

Actions Provided for Nodes , continued

Action	Description	NNMi Role Default/Minimum	Default Object Access Privilege
	(*NNMi Advanced*) If the Global Network Management feature is enabled and you are signed into a Global Manager: • Node managed by the Global Manager = **Actions → Configuration Details → Communication Settings** opens a report, provided by the Global Manager (NNMi management server). • Node managed by a Regional Manager = **Actions → Configuration Details → Communication Settings** accesses that Regional Manager (NNMi management server) and requests the report. **Note:** You must sign into that Regional Manager unless your network environment enables Single Sign-On (SSO) to that Regional Manager through the Global Manager. For more information, see the "Configuring Single Sign-On for Global Network Management" section in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.		
Monitoring Settings	**Configuration Details → Monitoring Settings** Displays the Monitoring Settings report about a particular node's SNMP Agent. (*NNMi Advanced*) If the Global Network Management feature is enabled and you are signed into a Global Manager: • Node managed by the Global Manager = **Actions → Configuration Details → Monitoring Settings** opens a report, provided by the Global Manager (NNMi management server). • Node managed by a Regional Manager = **Actions → Configuration Details →**	Operator Level 1/ Operator Level 1	Object Operator Level 1

Actions Provided for Nodes , continued

Action	Description	NNMi Role Default/Minimum	Default Object Access Privilege
	Monitoring Settings accesses that Regional Manager (NNMi management server) and requests the report. **Note:** You must sign into that Regional Manager unless your network environment enables Single Sign-On (SSO) to that Regional Manager through the Global Manager. For more information, see the “Configuring Single Sign-On for Global Network Management” section in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.		
List Supported MIBs	**MIB Information → List Supported MIBs** Display a list of the MIBs (Management Information Base) supported by a selected node. See "Determine the MIBs Supported for a Node (for Administrators)" on page 1451 and Determine a Node's Supported MIBs (MIB Browser) for more information.	Operator Level 2/ Operator Level 1	Object Operator Level 2
Browse MIB	**MIB Information → Browse MIB** The MIB Browser displays the responses to NNMi's SNMP requests made to a particular node in your network environment.	Operator Level 2/ Operator Level 1	Object Operator Level 2
Node Group Membership	Create or modify a Node Group using the selected nodes. This action also enables you to remove Node Groups. See "Create Node Groups From the Actions Menu " on page 312	Operator Level 1/ Operator Level 1	Object Administrator
Custom Attributes	Add Custom Attributes to the selected nodes or interfaces. See "Add Custom Attributes to Multiple Nodes or Interfaces Using the Actions Menu" on page 484	Administrator/ Administrator	Object Administrator
Open from Regional Manager	Issues a request to the Regional Manager (the NNMi management server that is responsible for monitoring the selected node) asking to display the Node form of the selected object.	Operator Level 1/ Operator Level 1	Object Operator Level 1

Actions Provided for Nodes , continued

Action	Description	NNMi Role Default/Minimum	Default Object Access Privilege
	Note: You must sign into that Regional Manager unless your network environment enables Single Sign-On (SSO) to that Regional Manager through the Global Manager. For more information, see the "Configuring Single Sign-On for Global Network Management" section in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.		
Regional Manager Console	Issues a request to the Regional Manager (the NNMi management server that is responsible for monitoring the selected node) asking to display the NNMi console. **Note**: You must sign into that Regional Manager unless your network environment provides Single Sign-On (SSO) to that Regional Manager.	Operator Level 1/ Operator Level 1	Object Operator Level 1
Delete	Deletes the selected object or objects. (*NNMi Advanced*) If the Global Network Management feature is enabled and you are signed into a Global Manager: • Node managed by the Global Manager = Actions → Delete removes the Node object (and all related object data) from the Global Manager's database. • Node managed by a Regional Manager = Actions → Delete removes the *copy of the Node object* (and all related object data) from the Global Manager's database. **Note:** To delete this Node object from the Regional Manager's database, click Actions → Open from Regional Manager and delete the Node object. You must sign into that Regional Manager unless your network environment enables Single Sign-On (SSO)	Administrator/ Administrator	Object Administrator

Actions Provided for Nodes , continued

Action	Description	NNMi Role Default/Minimum	Default Object Access Privilege
	to that Regional Manager through the Global Manager. For more information, see the “Configuring Single Sign-On for Global Network Management” section in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.		
Manage	**Management Mode → Manage** Changes the Management Mode of the selected node to **Managed**. Leaves the Direct Management Mode of any contained interfaces and addresses unchanged. (*NNMi Advanced*) If the Global Network Management feature is enabled and you are signed into a Global Manager: • Node managed by the Global Manager = Actions → Management Mode → Manage modifies the Node object in the Global Manager's database. • Node managed by a Regional Manager = Use Actions → Open from Regional Manager to set Management Mode on the Regional Manager that is responsible for this Node. **Note:** You must sign into that Regional Manager unless your network environment enables Single Sign-On (SSO) to that Regional Manager through the Global Manager. For more information, see the “Configuring Single Sign-On for Global Network Management” section in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.	Operator Level 2/ Operator Level 2	Object Operator Level 2
Manage	**Management Mode → Manage (Reset All)**	Operator Level	Object

Actions Provided for Nodes , continued

Action	Description	NNMi Role Default/Minimum	Default Object Access Privilege
(Reset All)	Changes the Management Mode of the selected node to **Managed**. Sets the Direct Management Mode of all contained interfaces and addresses to **Inherited**. (*NNMi Advanced*) If the Global Network Management feature is enabled and you are signed into a Global Manager: • Node managed by the Global Manager = Actions → Management Mode → Manage (Reset All) modifies the Node object plus all associated interface objects and address objects in the Global Manager's database. • Node managed by a Regional Manager = Use Actions → Open from Regional Manager to set Management Mode on the Regional Manager that is responsible for this Node. **Note:** You must sign into that Regional Manager unless your network environment enables Single Sign-On (SSO) to that Regional Manager through the Global Manager. For more information, see the “Configuring Single Sign-On for Global Network Management” section in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.	2/ Operator Level 2	Operator Level 2
Not Managed	**Management Mode → Not Managed** Changes the Management Mode of the node to **Not Managed**. Leaves the Direct Management Mode of any contained interfaces and addresses unchanged. (*NNMi Advanced*) If the Global Network Management feature is enabled and you are signed into a Global Manager: • Node managed by the Global Manager = Actions > Management Mode > Unmanage modifies the	Operator Level 2/ Operator Level 2	Object Operator Level 2

Actions Provided for Nodes , continued

Action	Description	NNMi Role Default/Minimum	Default Object Access Privilege
	Node object in the Global Manager's database. - Node managed by a Regional Manager = Use Actions > Open from Regional Manager to set Management Mode on the Regional Manager that is responsible for this Node. **Note:** You must sign into that Regional Manager unless your network environment enables Single Sign-On (SSO) to that Regional Manager through the Global Manager. For more information, see the "Configuring Single Sign-On for Global Network Management" section in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.		
Out of Service	**Management Mode → Out of Service** Changes the Management Mode of the selected node to **Out of Service**. Leaves the Direct Management Mode of any contained interfaces or addresses unchanged. (*NNMi Advanced*) If the Global Network Management feature is enabled and you are signed into a Global Manager: - Node managed by the Global Manager = Actions > Management Mode > Out of Service modifies the Node object in the Global Manager's database. - Node managed by a Regional Manager = Use Actions > Open from Regional Manager to set Management Mode on the Regional Manager that is responsible for this Node. **Note:** You must sign into that Regional Manager unless your network environment enables Single Sign-On (SSO) to that Regional Manager through the Global	Operator Level 2/ Operator Level 2	Object Operator Level 2

Actions Provided for Nodes , continued

Action	Description	NNMi Role Default/Minimum	Default Object Access Privilege
	Manager. For more information, see the "Configuring Single Sign-On for Global Network Management" section in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.		
Run Diagnostics (iSPI NET only)	(*HP Network Node Manager iSPI Network Engineering Toolset Software*) When installed, NNM iSPI NET gathers diagnostic information on the current node.	Operator Level 1/ Operator Level 1	Object Operator Level 1
Show Attached End Nodes	Displays information about the end nodes that NNMi determines are attached to the specified switch. (*NNMi Advanced*) If the Global Network Management feature is enabled and you are signed into a Global Manager: The results are based on the current information in the NNMi database of the Global Manager (which contains *copies of Node objects* from all Regional Managers).	Operator Level 1/ Operator Level 1	Object Operator Level 1

Actions Provided for Interfaces

Action	Description	NNMi Role Default/Minimum	Default Object Access Privilege
Graphs	Displays a pre-configured graph of real-time data for a selected interface. NNMi provides a set of Line Graphs that are configured to display real-time SNMP data. See Line Graphs Provided by NNMi for more information. Line Graphs can also come from the following sources: • Your NNMi administrator might configure additional graphs. • NNMi SPI software.	Operator Level 1/ Guest	Object Operator Level 1

Actions Provided for Interfaces , continued

Action	Description	NNMi Role Default/Minimum	Default Object Access Privilege
Status Poll	**Polling → Status Poll** Instructs NNMi to gather real-time data for all the information that NNMi uses to calculate Status for each selected Interface. A window for each Interface displays with a report about which information was gathered. The NNMi administrator determines the list of information gathered by establishing Monitoring configuration settings. See "Monitoring Network Health" on page 340 for more information. Note the following: • Status Poll might cause an object's Status to be updated.To see the resulting Interface Status, see Verify Current Status of a Device. • Using **Actions** → **Status Poll** does not affect the timing of the Polling interval configured for the device. **Tip**: The nnmstatuspoll.ovpl command line tool does the same thing as **Actions** → **Status Poll**. (*NNMi Advanced*) If the Global Network Management feature is enabled and you are signed into a Global Manager: • Node managed by the Global Manager = Actions → Status Poll requests that the Global Manager (NNMi management server) perform a status poll on the node. • Node managed by a Regional Manager = Actions → Status Poll requests that the Regional Manager perform a status poll on the node, the Global Manager displays the results. Latest Status Poll results are available on both NNMi management servers (Global and Regional). **Note:** You do not need to sign-in to the Regional Manager.	Operator Level 2/ Operator Level 2	Object Operator Level 2
Monitoring Settings	**Configuration Details** > **Monitoring Settings** Displays the Monitoring Settings report about a particular Interface.	Operator Level 1/ Operator Level 1	Object Operator Level 1

Actions Provided for Interfaces , continued

Action	Description	NNMi Role Default/Minimum	Default Object Access Privilege
Custom Attributes	Add Custom Attributes to the selected nodes or interfaces. See "Add Custom Attributes to Multiple Nodes or Interfaces Using the Actions Menu" on page 484	Administrator/ Administrator	Object Administrator
Manage	**Management Mode → Manage** Changes the Direct Management Mode of the interface to **Inherited**. Leaves the Direct Management Mode of any associated addresses unchanged. (*NNMi Advanced*) If the Global Network Management feature is enabled and you are signed into a Global Manager: • Node managed by the Global Manager = Actions → Management Mode → Manage modifies the Node object in the Global Manager's database. • Node managed by a Regional Manager = Use Actions → Open from Regional Manager to set Management Mode on the Regional Manager that is responsible for this Node. **Note:** You must sign into that Regional Manager unless your network environment enables Single Sign-On (SSO) to that Regional Manager through the Global Manager. For more information, see the “Configuring Single Sign-On for Global Network Management” section in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.	Operator Level 2/ Operator Level 2	Object Operator Level 2
Manage (Reset All)	**Management Mode → Manage (Reset All)** Changes the Management Mode of the interface to **Inherited**. Changes the Direct Management Mode of any associated addresses to **Inherited**. (*NNMi Advanced*) If the Global Network Management feature is enabled and you are signed into a Global Manager: • Node managed by the Global Manager = Actions →	Operator Level 2/ Operator Level 2	Object Operator Level 2

Actions Provided for Interfaces , continued

Action	Description	NNMi Role Default/Minimum	Default Object Access Privilege
	Management Mode → Manage (Reset All) modifies the Node object plus all associated interface objects and address objects in the Global Manager's database. • Node managed by a Regional Manager = Use Actions → Open from Regional Manager to set Management Mode on the Regional Manager that is responsible for this Node. **Note:** You must sign into that Regional Manager unless your network environment enables Single Sign-On (SSO) to that Regional Manager through the Global Manager. For more information, see the “Configuring Single Sign-On for Global Network Management” section in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.		
Not Managed	**Management Mode → Not Managed** Changes the Management Mode of the interface to Not Managed. Leaves the Direct Management Mode of any associated addresses unchanged. (*NNMi Advanced*) If the Global Network Management feature is enabled and you are signed into a Global Manager: • Node managed by the Global Manager = Actions > Management Mode > Unmanage modifies the Node object in the Global Manager's database. • Node managed by a Regional Manager = Use Actions > Open from Regional Manager to set Management Mode on the Regional Manager that is responsible for this Node. **Note:** You must sign into that Regional Manager unless your network environment enables Single Sign-On (SSO) to that Regional Manager through the Global	Operator Level 2/ Operator Level 2	Object Operator Level 2

Actions Provided for Interfaces , continued

Action	Description	NNMi Role Default/Minimum	Default Object Access Privilege
	Manager. For more information, see the "Configuring Single Sign-On for Global Network Management" section in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.		
Out of Service	**Management Mode → Out of Service** Changes the Management Mode of the interface to **Out of Service**. (*NNMi Advanced*) If the Global Network Management feature is enabled and you are signed into a Global Manager: • Node managed by the Global Manager = Actions > Management Mode > Out of Service modifies the Node object in the Global Manager's database. • Node managed by a Regional Manager = Use Actions > Open from Regional Manager to set Management Mode on the Regional Manager that is responsible for this Node. **Note:** You must sign into that Regional Manager unless your network environment enables Single Sign-On (SSO) to that Regional Manager through the Global Manager. For more information, see the "Configuring Single Sign-On for Global Network Management" section in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.	Operator Level 2/ Operator Level 2	Object Operator Level 2

Actions Provided for Addresses

Action	Description	NNMi Role Default/Minimum	Default Object Access Privilege
Layer 2 Neighbor View	**Maps** > **Layer 2 Neighbor View** Represents your network's physical connections and LAN switch traffic routes.	Operator Level 1/ Guest	Object Operator Level 1
Layer 3 Neighbor View	**Maps** > **Layer 3 Neighbor View** Represents your network's router traffic.	Operator Level 1/ Guest	Object Operator Level 1
Node Group Map	**Maps** → **Node Group Map** Displays the lowest level Node Group map to which the Node that is hosting the selected IP Address belongs. For example, if the Node belongs to a *Child* Node Group, the *Child* Node Group displays. See Node Group Maps. If the Node that is hosting the selected IP address belongs to multiple Node Groups, NNMi displays the list of possible Node Groups. Right-click the Node Group of interest and select **Maps** > **Node Group Map**. When the selected Source Node is not a member of any Node Group, and you select the **Node Group Map** action, NNMi displays an information message.	Operator Level 1/ Operator Level 1	Object Operator Level 1
Path View	Displays a map showing the route between two specified nodes, using the selected IP Address as the starting point. **Note**: *NNMi Advanced*. Path View works only with IPv4 addresses. The NNMi Advanced IPv6 address values are not valid choices for Path View. Any devices in your network that are configured with IPv6 addresses cannot be displayed on Path View maps.	Operator Level 1/ Guest	Object Operator Level 1
Ping (from server)	**Node Access** → **Ping (from server)** Tests whether the Node that is hosting the selected IP Address is reachable using the ping command. (*NNMi Advanced*) If the Global Network Management feature is enabled and you are signed into a Global Manager:	Operator Level 1/ Operator Level 1	Object Operator Level 1

Actions Provided for Addresses , continued

Action	Description	NNMi Role Default/Minimum	Default Object Access Privilege
	• Node managed by the Global Manager = Actions → Ping issues an ICMP request from the Global Manager (NNMi management server). • Node managed by a Regional Manager = Actions → Ping accesses that Regional Manager (NNMi management server) and issues the ICMP request. **Note:** You must sign into that Regional Manager unless your network environment enables Single Sign-On (SSO) to that Regional Manager through the Global Manager. For more information, see the "Configuring Single Sign-On for Global Network Management" section in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.		
Open Web Page	Opens the default Web page for the selected IP Address.	Administrator/ Administrator	
Trace Route (from sever)	**Node Access → Trace Route (from sever)** Traces a route path using the traceroute command. (*NNMi Advanced*) If the Global Network Management feature is enabled and you are signed into a Global Manager: • Node managed by the Global Manager = **Actions → Node Access → Trace Route** issues a request from the Global Manager (NNMi management server). • Node managed by a Regional Manager = **Actions → Node Access → Trace Route** accesses that Regional Manager (NNMi management server) and issues the request in a manner appropriate for the operating system in use on the Regional Manager.	Operator Level 1/ Operator Level 1	

Actions Provided for Addresses , continued

Action	Description	NNMi Role Default/Minimum	Default Object Access Privilege
	Note: You must sign into that Regional Manager unless your network environment enables Single Sign-On (SSO) to that Regional Manager through the Global Manager. For more information, see the “Configuring Single Sign-On for Global Network Management” section in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.		
Telnet (from client)	**Node Access → Telnet (from client)** Uses Transmission Control Protocol (TCP) protocol from the computer that launched your current browser (not the NNMi management server) to open a Telnet (teletype network) virtual terminal command-line interface from the selected IP Address. See Establish Contact with a Node.	Operator Level 2/ Operator Level 2	Object Operator Level 2
Secure Shell (from client)	**Node Access → Secure Shell (from client)** Uses Secure Shell (SSH) protocol from the computer that launched your current browser (not the NNMi management server) to open a Secure Shell virtual terminal command-line interface from the selected IP Address. See Establish Contact with a Node.	Operator Level 2/ Operator Level 2	Object Operator Level 2
Communication Settings	**Configuration Details** > **Communication Settings** Displays the communication configuration information for the Node that is hosting the selected IP Address. (*NNMi Advanced*) If the Global Network Management feature is enabled and you are signed into a Global Manager: • Node managed by the Global Manager = **Actions → Configuration Details → Communication Settings** opens a report, provided by the Global Manager (NNMi	Administrator/ Administrator	Object Administrator

Actions Provided for Addresses , continued

Action	Description	NNMi Role Default/Minimum	Default Object Access Privilege
	management server). • Node managed by a Regional Manager = **Actions** → **Configuration Details** → **Communication Settings** accesses that Regional Manager (NNMi management server) and requests the report. **Note:** You must sign into that Regional Manager unless your network environment enables Single Sign-On (SSO) to that Regional Manager through the Global Manager. For more information, see the “Configuring Single Sign-On for Global Network Management” section in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.		
Monitoring Settings	**Configuration Details** > **Monitoring Settings** Displays the Monitoring Settings report about a particular IP address. (*NNMi Advanced*) If the Global Network Management feature is enabled and you are signed into a Global Manager: • Node managed by the Global Manager = **Actions** → **Configuration Details** → **Monitoring Settings** opens a report, provided by the Global Manager (NNMi management server). • Node managed by a Regional Manager = **Actions** → **Configuration Details** → **Monitoring Settings** accesses that Regional Manager (NNMi management server) and requests the report. **Note:** You must sign into that Regional Manager unless your network environment	Operator Level 1/ Operator Level 1	Object Operator Level 1

Actions Provided for Addresses , continued

Action	Description	NNMi Role Default/Minimum	Default Object Access Privilege
	enables Single Sign-On (SSO) to that Regional Manager through the Global Manager. For more information, see the “Configuring Single Sign-On for Global Network Management” section in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.		
Status Poll	**Polling → Status Poll** Instructs NNMi to gather real-time data for all the information that NNMi uses to calculate Status for the Node that is hosting the selected IP Address. A window for each IP Address displays with a report about which information was gathered. The NNMi administrator determines the list of information gathered by establishing Monitoring configuration settings. See "Monitoring Network Health" on page 340 for more information.	Operator Level 2/ Operator Level 2	Object Operator Level 2
Configuration Poll	**Polling → Configuration Poll** Runs a real-time configuration check of the Node that is hosting the selected IP Address to detect any changes since the last discovery cycle. (*NNMi Advanced*) If the Global Network Management feature is enabled and you are signed into a Global Manager: • Node managed by the Global Manager = **Actions → Polling → Configuration Poll** results are provided by the Global Manager (NNMi management server). • Node managed by a Regional Manager = **Actions → Polling → Configuration Poll** requests an updated *copy* of the configuration information from the Regional Manager, then the Global Manager displays the results.	Operator Level 2/ Operator Level 2	Object Operator Level 2

Actions Provided for Addresses , continued

Action	Description	NNMi Role Default/Minimum	Default Object Access Privilege
	Note: You do not need to sign-in to the Regional Manager.		
Manage	**Management Mode → Manage** Changes the Direct Management Mode of the address to **Inherited**. (*NNMi Advanced*) If the Global Network Management feature is enabled and you are signed into a Global Manager: • Node managed by the Global Manager = Actions → Management Mode → Manage modifies the Node object in the Global Manager's database. • Node managed by a Regional Manager = Use Actions → Open from Regional Manager to set Management Mode on the Regional Manager that is responsible for this Node. **Note:** You must sign into that Regional Manager unless your network environment enables Single Sign-On (SSO) to that Regional Manager through the Global Manager. For more information, see the “Configuring Single Sign-On for Global Network Management” section in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.	Operator Level 2/ Operator Level 2	Object Operator Level 2
Not Managed	**Management Mode → Not Managed** Changes the management mode of the address to **Not Managed**. (*NNMi Advanced*) If the Global Network Management feature is enabled and you are signed into a Global Manager: • Node managed by the Global Manager = Actions > Management Mode > Unmanage modifies the	Operator Level 2/ Operator Level 2	Object Operator Level 2

Actions Provided for Addresses , continued

Action	Description	NNMi Role Default/Minimum	Default Object Access Privilege
	Node object in the Global Manager's database. • Node managed by a Regional Manager = Use Actions > Open from Regional Manager to set Management Mode on the Regional Manager that is responsible for this Node. **Note:** You must sign into that Regional Manager unless your network environment enables Single Sign-On (SSO) to that Regional Manager through the Global Manager. For more information, see the “Configuring Single Sign-On for Global Network Management” section in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.		
Out of Service	**Management Mode → Out of Service** Changes the management mode of the address to **Out of Service**. (*NNMi Advanced*) If the Global Network Management feature is enabled and you are signed into a Global Manager: • Node managed by the Global Manager = Actions > Management Mode > Out of Service modifies the Node object in the Global Manager's database. • Node managed by a Regional Manager = Use Actions > Open from Regional Manager to set Management Mode on the Regional Manager that is responsible for this Node. **Note:** You must sign into that Regional Manager unless your network environment enables Single Sign-On (SSO) to that Regional Manager through the Global Manager. For more information, see the “Configuring Single Sign-On for Global	Operator Level 2/ Operator Level 2	Object Operator Level 2

Actions Provided for Addresses , continued

Action	Description	NNMi Role Default/Minimum	Default Object Access Privilege
	Network Management” section in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.		

Actions Provided for Cards

Action	Description	NNMi Role Default/Minimum	Default Object Access Privilege
Status Poll	**Polling → Status Poll** Instructs NNMi to gather real-time data for all the information that NNMi uses to calculate Status for each selected card (maximum 10). A window for each card displays with a report about which information was gathered. The NNMi administrator determines the list of information gathered by establishing Monitoring configuration settings. See "Monitoring Network Health" on page 340 for more information.	Operator Level 2/ Operator Level 2	Object Operator Level 2
Monitoring Settings	**Configuration Details → Monitoring Settings** Displays the Monitoring Settings report about a particular card. (*NNMi Advanced*) If the Global Network Management feature is enabled and you are signed into a Global Manager: • Node managed by the Global Manager = **Actions → Configuration Details → Monitoring Settings** opens a report, provided by the Global Manager (NNMi management server). • Node managed by a Regional Manager = **Actions → Configuration Details → Monitoring Settings** accesses that Regional Manager (NNMi management server) and requests the report.	Operator Level 1/ Operator Level 1	Object Operator Level 1

Actions Provided for Cards, continued

Action	Description	NNMi Role Default/Minimum	Default Object Access Privilege
	Note: You must sign into that Regional Manager unless your network environment enables Single Sign-On (SSO) to that Regional Manager through the Global Manager. For more information, see the "Configuring Single Sign-On for Global Network Management" section in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.		
Manage	**Management Mode → Manage** Changes the Direct Management Mode of the card to **Inherited**. (*NNMi Advanced*) If the Global Network Management feature is enabled and you are signed into a Global Manager: • Node managed by the Global Manager = Actions → Management Mode → Manage modifies the Node object in the Global Manager's database. • Node managed by a Regional Manager = Use Actions → Open from Regional Manager to set Management Mode on the Regional Manager that is responsible for this Node. **Note:** You must sign into that Regional Manager unless your network environment enables Single Sign-On (SSO) to that Regional Manager through the Global Manager. For more information, see the "Configuring Single Sign-On for Global Network Management" section in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.	Operator Level 2/ Operator Level 2	Object Operator Level 2
Not Managed	**Management Mode** > **Not Managed** Changes the management mode of the card to **Not Managed**.	Operator Level 2/ Operator Level 2	Object Operator Level

Actions Provided for Cards, continued

Action	Description	NNMi Role Default/Minimum	Default Object Access Privilege
	(*NNMi Advanced*) If the Global Network Management feature is enabled and you are signed into a Global Manager: • Node managed by the Global Manager = Actions > Management Mode > Unmanage modifies the Node object in the Global Manager's database. • Node managed by a Regional Manager = Use Actions > Open from Regional Manager to set Management Mode on the Regional Manager that is responsible for this Node. **Note:** You must sign into that Regional Manager unless your network environment enables Single Sign-On (SSO) to that Regional Manager through the Global Manager. For more information, see the “Configuring Single Sign-On for Global Network Management” section in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.		2
Out of Service	**Management Mode** > **Out of Service** Changes the management mode of the card to **Out of Service**. (*NNMi Advanced*) If the Global Network Management feature is enabled and you are signed into a Global Manager: • Node managed by the Global Manager = Actions > Management Mode > Out of Service modifies the Node object in the Global Manager's database. • Node managed by a Regional Manager = Use Actions > Open from Regional Manager to set Management Mode on the Regional Manager that is responsible for this Node. **Note:** You must sign into that Regional Manager unless your network environment enables Single	Operator Level 2/ Operator Level 2	Object Operator Level 2

Actions Provided for Cards, continued

Action	Description	NNMi Role Default/Minimum	Default Object Access Privilege
	Sign-On (SSO) to that Regional Manager through the Global Manager. For more information, see the "Configuring Single Sign-On for Global Network Management" section in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.		

Actions Provided for Chassis

Action	Description	NNMi Role Default/Minimum	Default Object Access Privilege
Status Poll	**Polling → Status Poll** Instructs NNMi to gather real-time data for all the information that NNMi uses to calculate Status for each selected chassis (maximum 10). A window for each chassis displays with a report about which information was gathered. The NNMi administrator determines the list of information gathered by establishing Monitoring configuration settings. See "Monitoring Network Health" on page 340 for more information.	Operator Level 2/ Operator Level 2	Object Operator Level 2
Monitoring Settings	**Configuration Details → Monitoring Settings** Displays the Monitoring Settings report about a particular chassis. (*NNMi Advanced*) If the Global Network Management feature is enabled and you are signed into a Global Manager: • Node managed by the Global Manager = **Actions → Configuration Details → Monitoring Settings** opens a report, provided by the Global Manager (NNMi management server). • Node managed by a Regional Manager = **Actions → Configuration Details → Monitoring Settings** accesses that Regional Manager (NNMi	Operator Level 1/ Operator Level 1	Object Operator Level 1

Actions Provided for Chassis, continued

Action	Description	NNMi Role Default/Minimum	Default Object Access Privilege
	management server) and requests the report. **Note:** You must sign into that Regional Manager unless your network environment enables Single Sign-On (SSO) to that Regional Manager through the Global Manager. For more information, see the “Configuring Single Sign-On for Global Network Management” section in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.		
Manage	**Management Mode → Manage** Changes the Direct Management Mode of the chassis to **Inherited**. (*NNMi Advanced*) If the Global Network Management feature is enabled and you are signed into a Global Manager: • Node managed by the Global Manager = Actions → Management Mode → Manage modifies the Node object in the Global Manager's database. • Node managed by a Regional Manager = Use Actions → Open from Regional Manager to set Management Mode on the Regional Manager that is responsible for this Node. **Note:** You must sign into that Regional Manager unless your network environment enables Single Sign-On (SSO) to that Regional Manager through the Global Manager. For more information, see the “Configuring Single Sign-On for Global Network Management” section in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.	Operator Level 2/ Operator Level 2	Object Operator Level 2
Not Manage	**Management Mode** > **Not Managed**	Operator Level 2/ Operator	Object Operato

Actions Provided for Chassis, continued

Action	Description	NNMi Role Default/Minimum	Default Object Access Privilege
d	Changes the management mode of the chassis to **Not Managed**. (*NNMi Advanced*) If the Global Network Management feature is enabled and you are signed into a Global Manager: • Node managed by the Global Manager = Actions > Management Mode > Unmanage modifies the Node object in the Global Manager's database. • Node managed by a Regional Manager = Use Actions > Open from Regional Manager to set Management Mode on the Regional Manager that is responsible for this Node. **Note:** You must sign into that Regional Manager unless your network environment enables Single Sign-On (SSO) to that Regional Manager through the Global Manager. For more information, see the “Configuring Single Sign-On for Global Network Management” section in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.	Level 2	r Level 2
Out of Service	**Management Mode** > **Out of Service** Changes the management mode of the chassis to **Out of Service**. (*NNMi Advanced*) If the Global Network Management feature is enabled and you are signed into a Global Manager: • Node managed by the Global Manager = Actions > Management Mode > Out of Service modifies the Node object in the Global Manager's database. • Node managed by a Regional Manager = Use Actions > Open from Regional Manager to set Management Mode on the Regional Manager that is responsible for this Node.	Operator Level 2/ Operator Level 2	Object Operator Level 2

Actions Provided for Chassis, continued

Action	Description	NNMi Role Default/Minimum	Default Object Access Privilege
	Note: You must sign into that Regional Manager unless your network environment enables Single Sign-On (SSO) to that Regional Manager through the Global Manager. For more information, see the "Configuring Single Sign-On for Global Network Management" section in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.		

Actions Provided for Node Groups

Action	Description	NNMi Role Default/Minimum	Default Object Access Privilege
Node Group Map	**Maps** > **Node Group Map** Displays a current map of all nodes that belong to the selected Node Group.	Operator Level 1/ Operator Level 1	Object Operator Level 1
Preview Members	**Node Group Details > Preview Members (Current Group Only)** Displays a list of all nodes that belong to the selected Node Group as well as all of its Child Node Groups.	Operator Level 1/ Operator Level 1	Object Operator Level 1
Show Members	**Node Group Details > Show Members (Include Child Groups)** Displays a list of all nodes that belong to the selected Node Group.	Operator Level 1/ Operator Level 1	Object Operator Level 1
Show All Incidents	**Node Group Details** > **Show All Incidents** Checks for any Incidents associated with the selected Node Group.	Operator Level 1/ Operator Level 1	Object Operator Level 1
Show All Open Incidents	**Node Group Details** > **Show All Open Incidents** Checks for any open Incidents associated with the selected Node Group.	Operator Level 1/ Operator Level 1	Object Operator Level 1

Actions Provided for Node Groups , continued

Action	Description	NNMi Role Default/Minimum	Default Object Access Privilege
Status Details	**Node Group Details** > **Status Details** Displays a report about the status of all members of the selected Node Group. See Check Status Details for a Node Group.	Operator Level 1/ Operator Level 1	Object Operator Level 1

Actions Provided for Interface Groups

Action	Description	NNMi Role Default/Minimum	Default Object Access Privilege
Show Members	**Show Members** Displays a list of all nodes that belong to the selected Node Group.	Operator Level 1/ Operator Level 1	Object Operator Level 1

Actions Provided for Router Redundancy Groups

Action	Description	NNMi Role Default/Minimum	Default Object Access Privilege
Ping	**Node Access** > **Ping** Tests whether the node is reachable using the ping command from the NNMi console.	Operator Level 1/ Operator Level 1	Object Operator Level 1

Actions Provided for Router Redundancy Group Members

Action	Description	NNMi Role Default/Minimum	Default Object Access Privilege
Monitoring Settings	**Configuration Details** → **Monitoring Settings** Displays the Monitoring Settings report about a particular Router Redundancy Member.	Operator Level 1/ Operator Level 1	Object Operator Level 1

Actions Provided for Custom Polled Instances

Action	Description	NNMi Role Default/Minimum	Default Object Access Privilege
Graph Polled Instance	Graphs the line representing the Custom Poll results for the selected Custom Polled Instance. See Display an Line Graph for a Custom Polled Instance.	Operator Level 1/ Operator Level 1	Object Operator Level 1

Actions Provided for Custom Poller Collections and Report Groups (NNM iSPI Performance for Metrics only)

Action	Description	NNMi Role Default/Minimum	Default Object Access Privilege
Show Report Configuration	Displays the Report Collection configuration associated with the selected Custom Poller Collection or Report Group. See Create a Report Group and Create a Report Collection for more information..	Administrator/ Administrator	Object Administrator

Actions Provided for Node Components

Action	Description	NNMi Role Default/Minimum	Default Object Access Privilege
Status Poll	**Polling** > **Status Poll** Instructs NNMi to gather real-time data for all the information that NNMi uses to calculate Status for each selected Node Component. A window for each selected Node Component displays with a report about which information was gathered. The NNMi administrator determines the list of information gathered by establishing Monitoring configuration settings. See "Monitoring Network Health" on page 340 for more information.	Operator Level 2/ Operator Level 2	
Monitoring Settings	**Configuration Details** > **Monitoring Settings** Displays the Monitoring Settings report about the Node Component. See "Verify the Monitoring Settings" on page 416 and View the Monitoring Configuration Details.	Operator Level 1/ Operator Level 1	Object Operator Level 1
Manage	**Management Mode** > **Manage** Changes the Direct Management Mode of the Node Component to **Inherited**.	Operator Level 2/ Operator Level 2	
Not Managed	Changes the Management Mode of the Node Component to **Not Managed**.	Operator Level 2/ Operator Level 2	

NNMi Processes and Services

NNMi is built on a group of processes and services. For information about each process or service, see the following:

- "About Each NNMi Process" below
- "About Each NNMi Service" on the next page

To verify that everything is running properly, you can use the ovstatus command:

- "Verify that NNMi Processes Are Running" below

About Each NNMi Process

HP Network Node Manager Processes

Process Name	Description
OVsPMD	The control process that manages all the other NNMi processes.
pmd	Event Post Master daemon. This process routes events from the producers to the consumers. Producers of events are NNM 6.x/7.x management stations and processes. Consumers of events are the event pipeline and third-party applications.
ovjboss	The process that controls the NNMi application server that contains all of the NNMi Services (see "About Each NNMi Service" on the next page for more information).
nnmaction	The process that controls the Action Server. The NNMi Action Server runs any actions configured for incidents. See "Configure an Action for an Incident" on page 748 for more information about incident actions. See also the nnmaction Reference Page for more information
nmsdbmgr	NMS Database Manager. Controls the NNMi embedded database, including periodic database connectivity testing.

Verify that NNMi Processes Are Running

After you install Network Node Manager, a group of processes run on the NNMi management server.

To verify that all NNMi processes are running, do one of the following:

1. Select **Tools** → **NNMi Status** to display a report.
2. At the command line, type: **ovstatus –c**

   See the ovstatus Reference Page for more information.

Review the list of processes to ensure that all are running. For more information about each process, see "About Each NNMi Process" above.

Stop or Start an NNMi Process

You can stop and start NNMi processes from the command line. See the ovstop and ovstart Reference Pages for more information.

> **Caution:** If your NNMi management server participates in a high availability (HA) environment, under certain circumstances, you should not use `ovstop` or `ovstart`. Before using either of these commands, see the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.

To stop or start an NNMi process:

At the command line, type the appropriate command (see "About Each NNMi Process" on the previous page):

`ovstop <process name>`

`ovstart <process name>`

Note: If you use `ovstop` and `ovstart` without providing a process name, NNMi stops and starts all NNMi processes.

To generate a list of process names, see "Verify that NNMi Processes Are Running" on the previous page.

About Each NNMi Service

NNMi Services run inside the ovjboss process. The ovjboss process controls the NNMi application server that contains all of the NNMi services.

HP Network Node Manager Services

ovjboss Service Name	Description
CommunicationModelService	Creates the cache for communication configuration and listens for changes.
CommunicationParametersStatsService	Tracks internal statistics for measuring SNMP and ICMP configuration performance.
CustomPoller	Provides MIB instance polling to augment out-of-the-box state polling (performed by StatePoller). Enables users to create configurations based on dynamic grouping. Data collected by CustomPoller can be consumed by the HP Network Node Manager iSPI Performance for Metrics Software.
IslandSpotterService	Automatically discovers any Island Node Groups using Layer 2 connectivity information in the topology. An Island Group is a group of fully-connected nodes discovered by NNMi, and NNMi determines this group is not connected to the rest of the topology.

HP Network Node Manager Services, continued

ovjboss Service Name	Description
ManagedNodeLicenseManager	Responsible for ensuring that the number of managed nodes does not exceed the NNMi licensed capacity limit.
MonitoringSettingsService	Calculates how to monitor each device based on the Monitoring Configuration settings.
NamedPoll	NMS Named Poll Service. Used to trigger immediate state polls for monitored objects. Used by the Causal Engine during neighbor analysis and interface up/down investigations.
NnmTrapService	Used by trapd to receive traps from the standalone Operating System TrapReceiver process and forwards them to events.
NmsApa	NMS Active Problem Analyzer (APA) service determines the root cause of network problems and reports the root cause to the NMS Event Service. The Causal Engine is a key component of the NNMi APA service.
NmsCustomCorrelation	Custom Correlation Service. Enables the NNMi administrator to correlate one or more child incidents under an existing incident or a new parent incident.
NmsDisco	NMS Discovery Service. Adds new devices to the database and keeps the configuration of the managed devices up to date in the database by periodically rechecking the configuration of the devices. State Poller uses the Discovery service results to determine what to monitor. The Causal Engine depends on the Discovery service to monitor node configurations. The Causal Engine uses the configuration information when calculating status and root cause. NNMi uses the information provided by the Discovery service to maintain current device configuration information.
NmsEvents	NMS Events Service. Populates and manages the information displayed in the incident table. The information displayed comes from the other NNMi services that are running on your system. The incidents are filtered so you see only the most important information about your network.

HP Network Node Manager Services, continued

ovjboss Service Name	Description
NmsEventsConfiguration	Handles incident configuration changes.
NmsExtensionNotificationService	Responsible for applications that are integrated into NNM using the extension deployment model.
NmsTrapReceiver	Used by NNMi events to receives traps from the NnmTrapService and sends them to the events pipeline.
PerformanceSpiConsumptionManager	Verifies licensing capacity for HP Network Node Manager iSPI Performance for Metrics Software.
SpmdjbossStart	The SpmdjbossStart service interacts with the OVsPMD process during startup (ovstart), shutdown (ovstop), and reporting on the status of the ovjboss services (ovstatus –v ovjboss). **Caution:** If your NNMi management server participates in a high availability (HA) environment, under certain circumstances, you should not use `ovstop` or `ovstart`. Before using either of these commands, see the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.
StagedIcmp	Used by the State Poller to ping IP addresses using the Internet Control Message Protocol (ICMP). Also used by auto-discovery if Ping Sweep is enabled.
StagedSnmp	Used by the State Poller and Discovery to perform Simple Network Management Protocol (SNMP) read-only queries.
StatePoller	NMS State Poller Service. State Poller collects measurements that assess the current state of discovered devices. This information is provided for the Causal Engine to use when calculating device health.
TrapConfigurationServices	Merges configuration changes between the NNMi database and Trap Server.
TrustManager	Manages the trust information that is used when making trust decisions. Decides whether credentials presented by a peer should be accepted.

HP Network Node Manager iSPI Network Engineering Toolset Software Services (*NNM iSPI NET*)

ovjboss Service Name	Description
RbaManager	Tracks internal statistics and provides performance counters related to diagnostic flow execution using HP Operations Orchestration servers through the HP Network Node Manager iSPI Network Engineering Toolset Software.

Verify that NNMi Services are Running

After you install Network Node Manager, a group of services run on the NNMi management server. For information about each service, see "About Each NNMi Service" on page 82.

To verify that all NNMi services are running, do one of the following:

- Select **Tools** → **NNMi Status** to display a report.
- At the command line, type:

  ovstatus –v ovjboss

  See the ovstatus Reference Page for more information.

Review the list of services to ensure that all are running.

"`Service is started`" means this service is working properly.

"`Service is stopped`" means this service/process is not running.

If you see any of the messages in this list, investigate the log files and look for the keyword **Exception** (within the log file for the parent `ovjboss` process and the log file for the specific service):

"Service is in created state"
"Service is in failed state"
"Service is in registered state"
"Service is in destroyed state"
"Service is in started state"
"Service is in starting state"
"Service is in stopped state"
"Service is in stopping state"
"Service is in unregistered state"

Log files are found in the following location. If your NNMi management server participates in a high availability (HA) environment, click here for more information:

1. Before opening the log file, first identify the `HA_MOUNT_POINT` for your NNMi environment.
2. At the command line, type:

   Windows:

   ```
   %NnmInstallDir%/misc/nnm/ha/nnmhaclusterinfo.ovpl NNM -config -get
   HA_MOUNT_POINT
   ```

UNIX:
`opt/OV/misc/nnm/ha/nnmhaclusterinfo.ovpl NNM -config -get HA_MOUNT_POINT`

3. At the command line, type the following (`/DataDir/` is the literal path):

   `<HA_MOUNT_POINT>/DataDir/log/nnm`

- **Windows:**
  `%NnmDataDir%\log\nnm\<name>.`**`%g`**
- **UNIX:**
  `/var/opt/OV/log/nnm/<name>.`**`%g.`**

%g represents the archive number of the archived log file

The parent ovjboss process generates the following log files:

- `ovjboss.log` and `ovjboss.log.old`

Note: Each restart creates a new `ovjboss.log` and overwrites the `ovjboss.log.old`.

Stop or Start NNMi Services

You can stop or start all NNMi services at the same time. You cannot start and stop individual services. See the ovstop and ovstart Reference Page for more information.

> **Caution:** If your NNMi management server participates in a high availability (HA) environment, under certain circumstances, you should not use `ovstop` or `ovstart`. Before using either of these commands, see the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.

To stop or start the NNMi services:

At the command line, type the command:

```
ovstop
```

```
ovstart
```

Chapter 2

Introduction to IPv6 (*NNMi Advanced*)

IPv6 upgrades IPv4 features and allows for vastly more address space, built-in security, and enhanced support for streaming media and peer-to-peer applications.

When your network environment includes both IPv4 and IPv6, your NNMi administrator can configure NNMi to automatically detect and monitor both types of addresses, whether devices are IPv4-only, IPv6-only, or dual-stack (both).

> **Note:** The NNMi administrator must enable IPv6 with a setting in the `nms-jboss.properties` file. See the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.

If enabled for IPv6, NNMi-Advanced allows IPv6 addresses and address ranges in the following:

- SNMP communication (address ranges and specific addresses), see "Configuring Communication Protocol" on page 119.

  The NNMi administrator specifies whether your network environment uses IPv4 or IPv6 addresses for SNMP agents Management Address (see "Configure Default SNMP, Management Address, and ICMP Settings" on page 120).

  Configuration workspace > **Communication Configuration** > in the **Management Address Selection**area, configure the **IP Version Preference** attribute (IPv4, IPv6, or Any)

- Discovering address information about devices in your network (Include or exclude address ranges and specific addresses. Use IPv6 for discovery seeds.). See "Discovering Your Network" on page 175 and "IPv6 Addresses Prerequisite (NNMi Advanced)" on page 190.

  > **Note:** IPv6 addresses are automatically excluded from Ping Sweep if you configure NNMi to use Ping Sweep as a starting point for discovery. NNMi detects the IPv6 addresses later in the discovery process. IPv6 addresses cannot be used when manually configuring representations of subnet connections that NNMi cannot otherwise detect (the optional Subnet Connection Rules aspect of Discovery).

- Monitoring a subset of the discovered devices using Node Group and Interface Group filters (through address filters and specific address lists), see "Monitoring Network Health" on page 340.
- NNMi uses ICMPv6 communication for IPv6 Address fault monitoring.
- Configure NNMi fault monitoring to generate incidents about a subset of the discovered devices using Node Group and Interface Group filters (address filters and specific addresses), see "Configuring Incidents" on page 589.

NNMi documents the associations between IPv6 Addresses, Subnets, Interfaces, and Nodes and presents consolidated IPv4 and IPv6 information. See Accessing Device Details.

NNMi provides Layer 2 Neighbor Views, Layer 3 Neighbor Views, and Topology Maps of IPv4 and IPv6 devices combined.

Note: Path View does not include IPv6 addresses.

The NNMi console's **Actions** → **Node Access** → **Ping (from server)** and **Actions** → **Node Access** → **Trace Route (from server)** menu items work with both IPv4 and IPv6. See Test Node Access (Ping) and Find the Route (traceroute).

Tip: You can right-click any object in a table or map view to access the **Actions** menu.

Chapter 3

Overlapping Addresses in NAT Environments

NNMi can help you manage areas in your network that are configured using address translation protocols, resulting in Overlapping Addresses / Duplicate Address Domains. Each address domain must be assigned to a unique Tenant, see "Configure Tenants" on page 194. Spiral Discovery requires a Discovery Seed (Tenant / Address pair) to identify each node before NNMi discovers and monitors that node. See "Specify Discovery Seeds" on page 256.

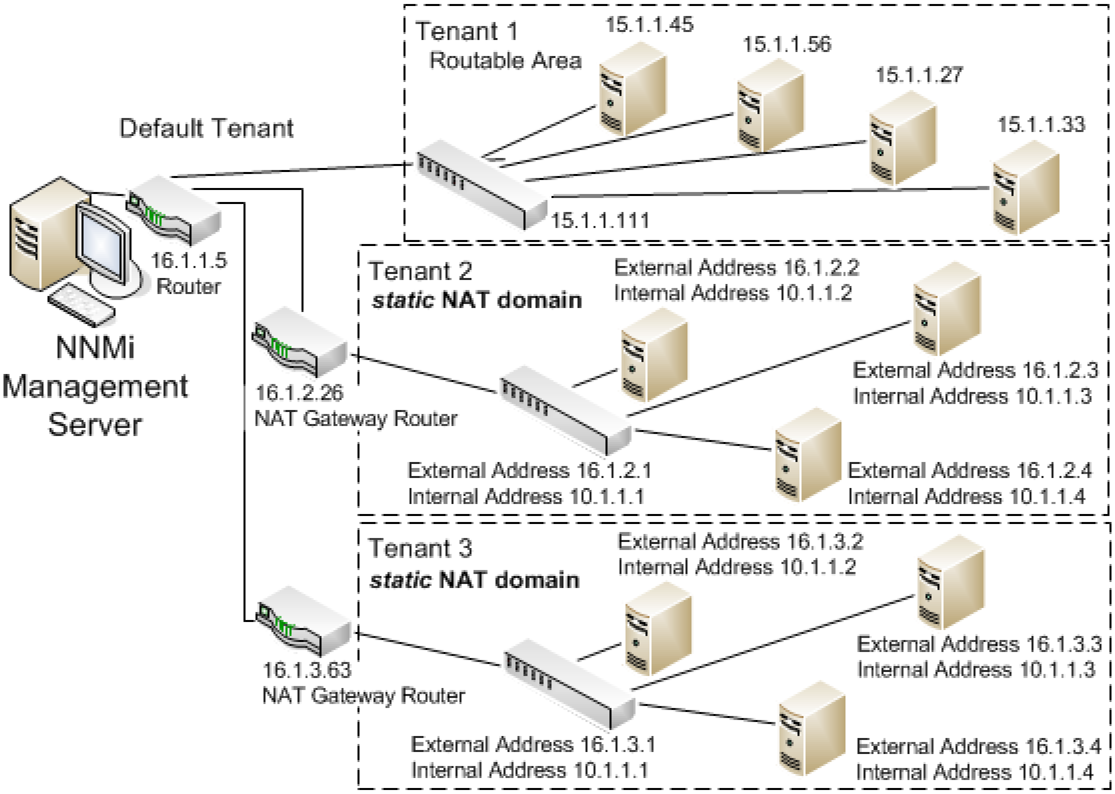
Best Practice: No duplicate Domain Name System (DNS) names. See "Well-Configured DNS Prerequisite" on page 188.

NNMi helps you manage important devices that are using any of the following address translation protocols. The NNMi configuration requirements vary, depending on the protocol:

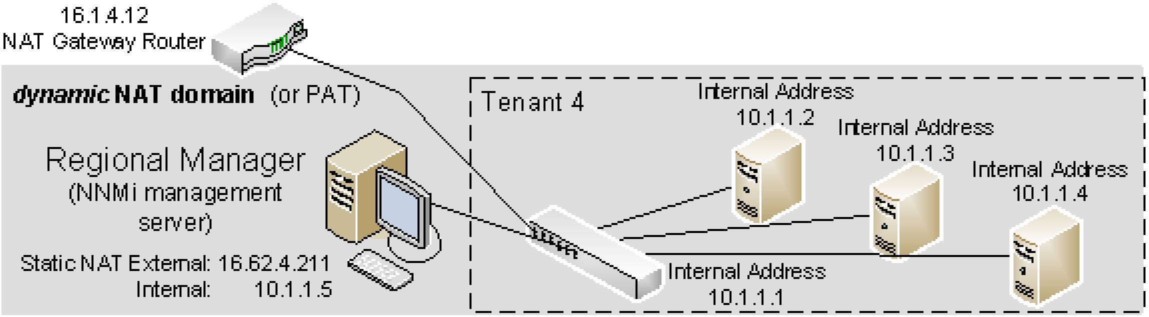
- *Static* Network Address Translation (NAT)
- *Dynamic* Network Address Translation (NAT)
- *Dynamic* Port Address Translation (PAT/NAPT)

One NNMi management server can manage one or more *static* Network Address Translation (NAT) domains, each address domain must be assigned to a unique Tenant.

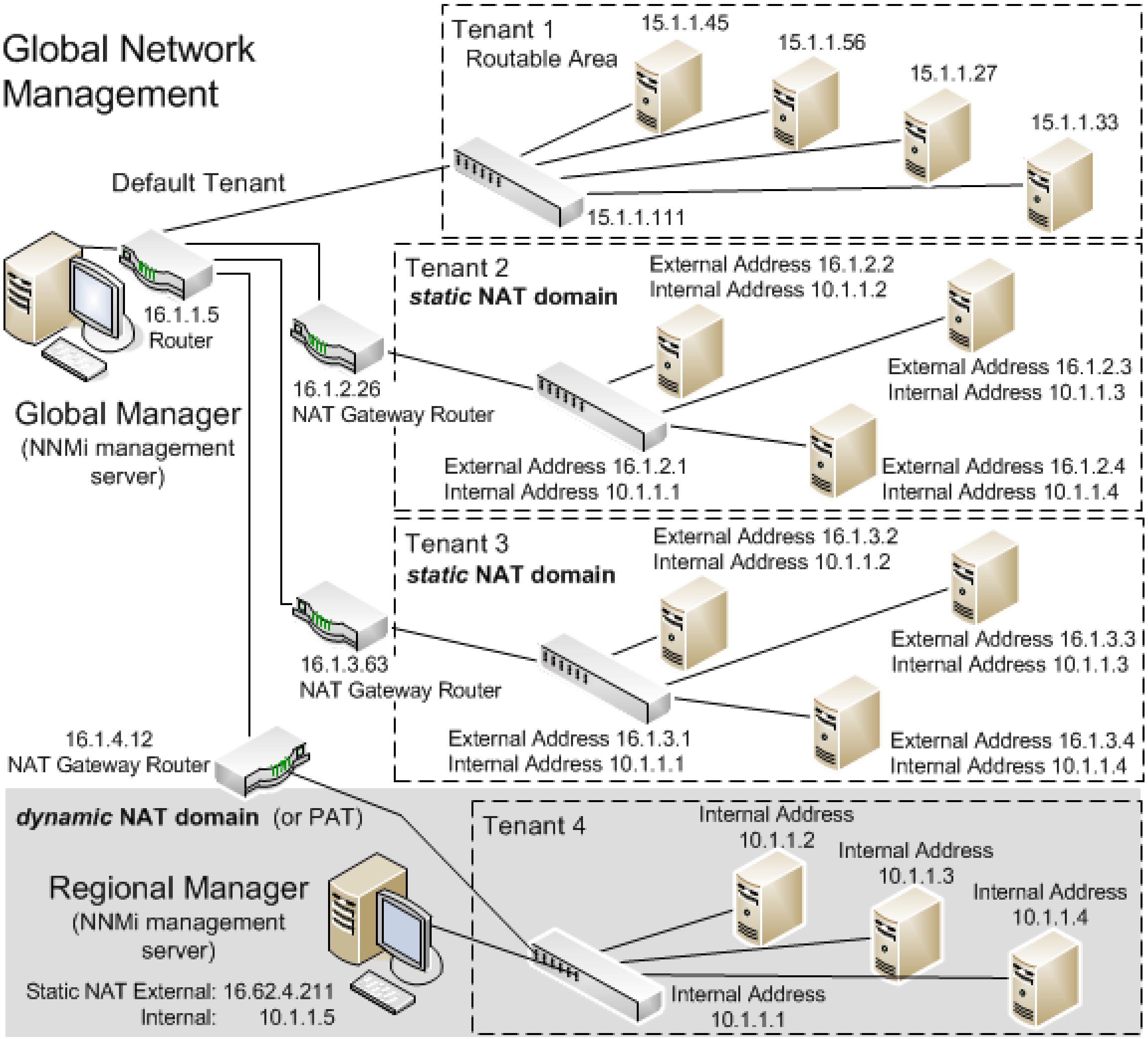
If *static* Network Address Translation (NAT) is part of your network management domain, you configure NNMi to display the NAT *external IP address* (public address) in the Mapped Address attribute of the IP Address form for the identified Tenant / NAT *internal IP address* (such as private IPv4 address) pair. This configuration setting is also important for node monitoring, see "Overlapping Address Mapping" on page 191.


Tenant 1
Routable Area
15.1.1.45
15.1.1.56
15.1.1.27
15.1.1.33
15.1.1.111
Default Tenant
16.1.1.5
Router
NNMi
Management
Server
Tenant 2
static NAT domain
External Address 16.1.2.2
Internal Address 10.1.1.2
External Address 16.1.2.3
Internal Address 10.1.1.3
16.1.2.26
NAT Gateway Router
External Address 16.1.2.1
Internal Address 10.1.1.1
External Address 16.1.2.4
Internal Address 10.1.1.4
Tenant 3
static NAT domain
External Address 16.1.3.2
Internal Address 10.1.1.2
External Address 16.1.3.3
Internal Address 10.1.1.3
16.1.3.63
NAT Gateway Router
External Address 16.1.3.1
Internal Address 10.1.1.1
External Address 16.1.3.4
Internal Address 10.1.1.4

One NNMi management server can manage one *dynamic* Network Address Translation (NAT) domain or *dynamic* Port Address Translation (PAT/NAPT) domain. All nodes in this domain must belong to the same Tenant. The NNMi management server must participate in a Global Network Management environment as a Regional Manager.



Use NNMi's Global Network Management feature to monitor multiple *dynamic* Network Address Translation (NAT), *dynamic* Port Address Translation (PAT/NAPT) domains, or both. Tenants must be unique within the entire NNMi Global Network Management configuration. See "Connecting Multiple NNMi Management Servers (NNMi Advanced)" on page 93 and "Tenant Best Practices for Global Network Management" on page 96.


Global Network Management
Default Tenant
16.1.1.5
Router
Global Manager
(NNMi management server)
Tenant 1
Routable Area
15.1.1.45
15.1.1.56
15.1.1.27
15.1.1.33
15.1.1.111
Tenant 2
static NAT domain
External Address 16.1.2.2
Internal Address 10.1.1.2
External Address 16.1.2.3
Internal Address 10.1.1.3
16.1.2.26
NAT Gateway Router
External Address 16.1.2.1
Internal Address 10.1.1.1
External Address 16.1.2.4
Internal Address 10.1.1.4
Tenant 3
static NAT domain
External Address 16.1.3.2
Internal Address 10.1.1.2
16.1.3.63
NAT Gateway Router
External Address 16.1.3.3
Internal Address 10.1.1.3
External Address 16.1.3.1
Internal Address 10.1.1.1
External Address 16.1.3.4
Internal Address 10.1.1.4
16.1.4.12
NAT Gateway Router
dynamic NAT domain (or PAT)
Regional Manager
(NNMi management server)
Static NAT External: 16.62.4.211
Internal: 10.1.1.5
Tenant 4
Internal Address 10.1.1.2
Internal Address 10.1.1.3
Internal Address 10.1.1.4
Internal Address 10.1.1.1

Devices that belong to the Default Tenant can have Layer 2 Connections to any device in any Tenant. Devices within any Tenant *other than* Default Tenant can have Layer 2 Connections *only* to devices within the same Tenant or the Default Tenant.

> **Tip:** Assign any infrastructure device that interconnects multiple NAT domains (such as a NAT gateway) to the Default Tenant. This ensures that NNMi displays the Layer 2 Connections your team and customers need to see.

For more information:

Chapter 4

Use NNMi Help Anywhere, Anytime

The NNMi Help system can run independently from the console. Simply unzip the files into any convenient location.

To locate the NNMi Help files, on the NNMi management server, navigate to the location appropriate for the NNMi management server's operating system (see table).

Location of the NNMi Help System

Operating System	NNMi Help System Files
Windows	`%NnmInstallDir%\NNM\server\deploy\nnmDocs_en.war`
UNIX	`/NNM/server/deploy/nnmDocs_en.war`

To access Help independently from the console:

1. Copy the web archive file `nnmDocs_en.war` to any convenient location.
2. At the command prompt, navigate to the directory where you placed the `nnmDocs_en.war` file. To extract the help directory structure and files, type:

   `jar xvf nnmDocs_en.war` (You might need to specify the complete path to the `jar` command's location on your computer.)

   Tip: You can also use WinZip on Windows to decompress the `nnmDocs_en.war` file.
3. Navigate to and open the `/htmlHelp/nmHelp/nmHelp.html` file.
4. The NNMi Help system runs as usual in the default browser window.

To Access a PDF version of the NNMi online help:

Go to: `http://h20230.www2.hp.com/selfsolve/manuals`

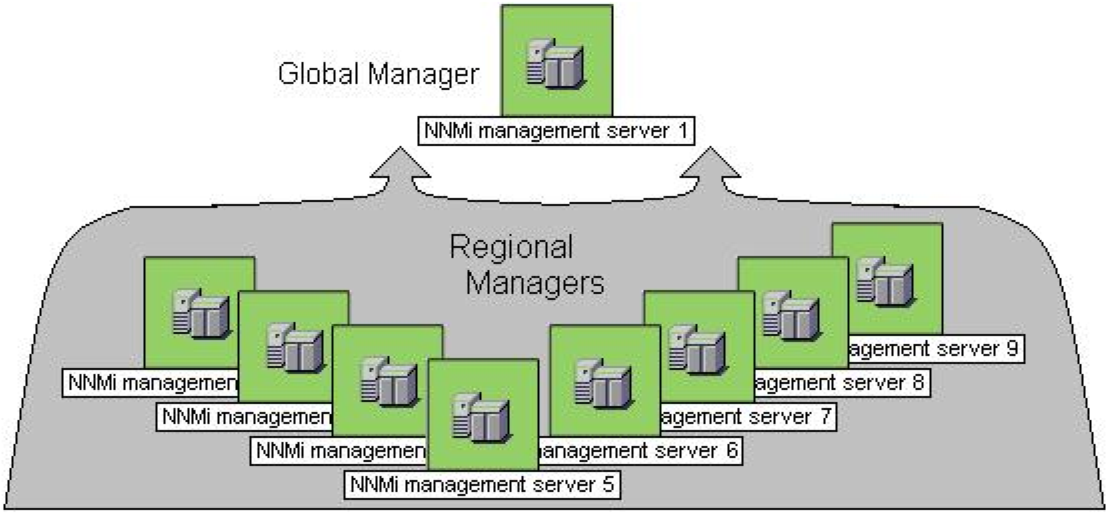
This site requires that you register for an HP Passport ID. To obtain an HP Passport ID, go to:

`http://h20229.www2.hp.com/passport-registration.html`

Chapter 5

Connecting Multiple NNMi Management Servers (*NNMi Advanced*)

The Global Network Management feature enables you to configure NNMi to share the workload among multiple NNMi management servers in your network environment. For more information about the Global Network Management feature, click here.



(*NNMi Advanced*) There are many benefits to using the NNMi Global Network Management feature:

- Provides safe and secure communication among multiple NNMi management servers.
- Provides a central big-picture view of your corporate-wide network on the Global Manager for 24-hour/7-days-per-week coverage.
- Enables management of nodes that are configured with address translation protocols to provide their public address (resulting in overlapping addresses domains). An NNMi Regional Manager is required for each address domain configured with following protocols:
  - *Static* Network Address Translation (NAT)
  - *Dynamic* Network Address Translation (NAT)
  - *Dynamic* Port Address Translation (PAT/NAPT)
- Easy to set up:
  - Each Regional Manager administrator specifies *all Node object data* or *a specific Node Group* for participation at the Global Manager level.
  - Each Global Manager administrator specifies which Regional Managers are permitted to contribute information.

- Automatically combines topology from multiple NNMi management servers on the Global Manager, but keeps management responsibilities separate. (No duplication, the responsible NNMi Management server is clearly identified per Node.)
- Generates and manages Incidents independently on each server (generated within the context of topology available on each server).
- Regional Manager administrators can configure specific SNMP traps or NNM 6.x/7.x Events to be forwarded from Regional Managers to Global Managers.

Review the Global Network Management deployment choices in the *HP Network Node Manager i Software Deployment Reference* (available at: `http://h20230.www2.hp.com/selfsolve/manuals`).

All NNMi management servers in your network environment that participate in Global Network Management (Global Managers and Regional Managers) or Single Sign-On (SSO) must have their internal time clocks synchronized in universal time.

> **Caution:** Use a Time Synchronization program, for example, the UNIX (HP-UX / Linux / Solaris) tool Network Time Protocol Daemon (NTPD) or one of the available Windows operating system tools.

Review the Global Network Management deployment choices and “Configuring Single Sign-On for Global Network Management” section in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.

To configure Global Network Management, do the following:

1. Navigate to the **Global Network Management** form.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Select **Global Network Management**.
2. Do one of the following:
   - **Global Manager**. If you are the NNMi administrator for the **Global Manager**, decide which Regional Managers are permitted to forward network information to that Global Manager (NNMi management server). Each Regional Manager retains management responsibilities for the forwarded nodes. The Global Manager might or might not directly manage a set of network devices. See the following topics for more information:
     - "Global Manager: Connect to a Regional Manager" on page 103
     - "Overlapping Addresses in NAT Environments" on page 89

     > **Note:** Each NNMi management server must have a static, routable IP address as the Management Address (for all SNMP/ICMP communication). See "Configure Default SNMP, Management Address, and ICMP Settings" on page 120 and "Specific Node Settings Form (Communication Settings)" on page 155.
   - **Regional Manager**. If you are the NNMi administrator for the **Regional Manager**, you control the following aspects of communication with the Global Manager:

- Configure a Tenant for each address domain. The Tenant name must be unique across all Tenants in the Global Network Management domain. See "Tenant Best Practices for Global Network Management" on the next page.
- Forward information about *all* Node object data or *only data about Nodes belonging to one Node Group*. See "Regional Manager: Create a Forwarding Filter (Limit the available Node information)" on page 102.

  > **Note:** Incidents associated with the specified Nodes are not forwarded to the Global Manager. *Each server maintains an independent group of incidents.*

- Forward specific SNMP traps and NNM 6.x/7.x Events to the Global Manager. See "Configure Forward to Global Manager Settings for an SNMP Trap Incident (NNMi Advanced)" on page 927 and "Configure Forward to Global Managers Settings for a Remote 6.x/7.x Event Incident (NNMi Advanced)" on page 1338.

3. Click **Save and Close** to apply your changes.

After Global Network Management is set up in your network environment:

- To troubleshoot any issued with Global Network Management, see "Troubleshoot Global Network Management" on page 110.
- To determine which Nodes are monitored by each NNMi management server, see View the NNMi Management Servers' Domain List.
- To determine which Incidents were forwarded to the Global Manager, see Monitor Incidents in a Global Network Management Environment (NNMi Advanced).

About Multi-Tenancy and Global Network Management

(*NNMi Advanced - Global Network Management feature*.) When configuring NNMi for multiple Tenants in a Global Network Management environment, note the following:

- All NNMi installations (NNMi Regional Managers and NNMi Global Managers) have a Tenant object named *Default* with the following UUID: 1b96011e-8829-4e5d-8ab7-f93b7b10ac79.
- If areas in your network are configured using address translation protocols, each address domain must be assigned to a unique Tenant.
- If a Regional Manager's Node is *replicated* to the Global Manager, and that *replicated Node* is assigned to a Tenant UUID that is not yet defined on the NNMi Global Manager, NNMi creates an additional Tenant definition on the NNMi Global Manager.

  Note: Ideally, this would never happen, see "Tenant Best Practices for Global Network Management" on the next page.

- If the NNMi Global Manager creates an additional Tenant object (based on a Regional Manager's replicated Node), NNMi uses the following attribute values for that new Tenant object:
  - **UUID**: The NNMi Global Manager uses the Regional Manager Tenant's *UUID* attribute value for the new Global Manager's Tenant definition.

- **Name**: The NNMi Global Manager automatically uses the Regional Manager Tenant's *Name* for that new Global Manager's Tenant object. However, the NNMi administrator on the Global Manager can change that name, but the UUID maintains the relationship. See "Troubleshooting Tenants in Global Network Management" on page 99.
- **Initial Discovery Security Group**: The NNMi Global Manager automatically creates a new Security Group with the same *Name* as that newly created Tenant, and uses that newly created Security Group for this attribute value.

  Note: The NNMi Global Manager creates this new Security Group whether a Security Group by that name already exists, and that duplicate will have a unique UUID. To avoid duplicates, see "Tenant Best Practices for Global Network Management" below.

  The NNMi Regional Manager Tenant's *Initial Discovery Security Group* attribute value is not preserved on the Global Manager because the Security configuration on the Global Manager represents the needs of a different network environment. By creating a new Security Group, no operators or guests on the NNMi Global Manager can see those replicated nodes unless an NNMi administrator intentionally creates an appropriate Security Group Mapping. See "Configuring Security" on page 503 for more information.

When additional Nodes from that Regional Manager are replicated to the NNMi Global Manager, for those Nodes, the NNMi Global Manager uses the same Tenant assigned by the Regional Manager (based on the *UUID* of the Tenant) and the *Initial Discovery Security Group* attribute value for that Tenant as defined on the Global Manager.

Tenants for Overlapping Address Domains

If your network uses any of the following address translation protocols, you must create a unique Tenant (other than *Default Tenant*) for each domain of nodes with addresses determined by the following protocols (see "Configure Tenants" on page 194):

- *Static* Network Address Translation (NAT)
- *Dynamic* Network Address Translation (NAT)
- *Dynamic* Port Address Translation (PAT/NAPT)

The configuration requirements vary, depending which protocol is used (see "Overlapping Addresses in NAT Environments" on page 89):

- Any number of *static* Network Address Translation (NAT) instances can be monitored by one NNMi management server, as long as each instance is configured with a unique Tenant.
- Each instance of *dynamic* Network Address Translation (NAT) or *dynamic* Port Address Translation (PAT/NAPT) must be configured as an NNMi Regional Manager, in addition to a unique Tenant. See "Connecting Multiple NNMi Management Servers (NNMi Advanced)" on page 93.

Tenant Best Practices for Global Network Management

NNMi Global Manager administrators and NNMi Regional Manager administrators need to *work together* to synchronize Tenants and Security Groups for replicated Nodes.

Note: If using HP Network Node Manager iSPI Performance for Metrics Software, HP Network Node Manager iSPI Performance for Quality Assurance Software, or HP Network Node Manager iSPI Performance for Traffic Software and you want to generate reports from the Global Manager, this Best Practice procedure is a required part of the configuration (not optional).

See also "About Multi-Tenancy and Global Network Management" on page 95 and "Troubleshooting Tenants in Global Network Management" on page 99.

Best practice procedure for establishing Tenants in a Global Network Management environment:

1. The NNMi administrators work together to agree on a *naming strategy* for the Tenants assigned to replicated Nodes and the Initial Discovery Security Group attribute value for those Tenants.

   When a Tenant is assigned to a particular Node, the associated Security Group for that Tenant can be different on the Regional Manager and Global Manager:

Node Attribute	Original Node's Attribute Value on NNMi's Regional Manager	Replicated Node's Attribute Value on NNMi's Global Manager
Tenant	Name: *ABC*	→ Same Name as Regional Setting.
Security Group	Name: < *strategy* > (These names can be independent of the Security Group names required by the Global Manager. Use any logic that works for your team.)	Name: < *strategy* > (These names can be independent of the Security Group names required by any of the Regional Managers. For example, consider names that indicate *which* Regional Manager replicated the Node.)

2. The NNMi Global Manager's administrator does the following according to the new naming strategy (determined in step 1):

   - Defines all Security Groups required by the Global Manager.

     If your team plans to use certain Security Groups on *multiple* NNMi management servers (Regional Managers / Global Manager), defines all those shared Security Groups. This establishes the UUID assigned to each shared Security Group.

   - Defines all Tenants required by the Regional Managers and all Tenants required by the Global Manager. This establishes the UUID assigned to each Tenant. For each Tenant's *Initial Discovery Security Group* attribute value, use one of the Security Groups that are appropriate for the Global Manager (because this setting is independent of the Regional Manager's setting).

   - Uses the `nnmconfigexport.ovpl` command line tool to *export* the new Tenant object definitions and Security Group object definitions for importing into each Regional Manager's database. See the nnmconfigexport.ovpl Reference Page.

   - Updates each Node's Tenant assignment (to match the naming strategy determined in step 1):

For *non-replicated* Nodes: Uses the `nnmsecurity.ovpl` command line tool to update Tenant assignments for each Node in the NNMi Global Manager's database to the newly created Tenants. See the nnmsecurity.ovpl Reference Page.

For *replicated* Nodes: After completing step 3, each *replicated* Node's *Tenant* assignment is automatically updated in the NNMi Global Manager's database (to match the Regional Manager's assignment the next time the Regional Manager forwards information about discovery and monitoring results to the Global Manager).

- Updates each Node's Security Group assignment (to match the naming strategy determined in step 1):

  Change existing Security Group assignments for *all* Nodes in the Global Manager's database using one of the following methods:

  - The Security Wizard. See "Using the Security Wizard View" on page 524.
  - The `nnmsecurity.ovpl` command line tool. See the nnmsecurity.ovpl Reference Page.

  Note: These Security Group assignments can be different from the Regional Manager's assignments, and any changes to the Regional Manager's Security Group assignment for each Node are not replicated from Regional Managers to the Global Manager.

3. Each Regional Manager's NNMi administrator does the following according to the new naming strategy (determined in step 1):

   - Uses the `nnmconfigimport.ovpl -c security` command line tool to import the new Tenant object definitions and Security Group object definitions (the Global Manager's exported settings). See the nnmconfigimport.ovpl Reference Page.
   - *Optional*. Deletes any imported Tenants that are not relevant for *this* Regional Manager.
   - If not using *shared* Security Groups: Modifies each Tenant's *Initial Discovery Security Group* setting to one of the Security Groups that are appropriate for *this* Regional Manager.
   - *Optional*. Deletes any imported Security Groups that are not relevant for *this* Regional Manager.
   - Updates each Node's Tenant assignment (to match the naming strategy determined in step 1):

     Use the `nnmsecurity.ovpl` command line tool to change each Node's *Tenant* assignment to the appropriate imported Tenant. See the nnmsecurity.ovpl Reference Page.
   - Updates each Node's Security Group assignment (to match the naming strategy determined in step 1):

     Change existing Security Group assignments for *all* Nodes in the Regional Manager's database using one of the following methods:

     - The Security Wizard. See "Using the Security Wizard View" on page 524.
     - The `nnmsecurity.ovpl` command line tool. See the nnmsecurity.ovpl Reference Page.

     Note: These Security Group assignments can be different from the Global Manager's assignments, and the changes to the Security Group assignments are not replicated to the Global Manager.

- Repeat step 3 for each Regional Manager.

Troubleshooting Tenants in Global Network Management

You need to understand how NNMi determines the Tenant and Security Group setting per replicated Node. For more information, see "About Multi-Tenancy and Global Network Management" on page 95.

The following scenarios explain the results of a potential series of changes when "Tenant Best Practices for Global Network Management" on page 96 was not followed:

1. The first time the Regional Manager forwards information about discovery and monitoring results to the Global Manager. Click here for details.

   When a Regional Manager's Nodes are assigned to a custom Tenant (other than *Default*) and those Nodes are replicated to the Global Manager, if the Global Manager's database does not contain a Tenant object with the same *UUID*:

   - The Global Manager creates a new Tenant object with the same UUID and Name.
   - The Global Manager automatically creates a new Security Group with the same *Name* as the Tenant. This happens whether a Security Group by that name already exists (the duplicate has a unique UUID).

Node Attribute	Original Node's Attribute Value on NNMi's Regional Manager	Replicated Node's Attribute Value on NNMi's Global Manager
Tenant	UUID: uniqueTenant#one Name: MyCustomer	→ Same UUID as Regional Setting. → Same Name as Regional Setting.
Security Group	UUID: uniqueSecurityGrp#one Name: Tier1Support	NNMi creates a new Security Group with same *Name* as the Regional Manager's custom Tenant name. All other attributes of this Security Group have no relation to the Regional Manager's Tenant object. UUID: *uniqueSecurityGrp#two* Name: *MyCustomer*

2. Regional Manager's NNMi administrator changes the name of the *MyCustomer* Tenant object. Click here for details.

   Changes to the NNMi Regional Manager's Tenant *Name* or *Description* are not replicated to the NNMi Global Manager. (No change on the Global Manager.)

Previously Replicated Node

Node Attribute	Original Node's Attribute Value on NNMi's Regional Manager	Replicated Node's Attribute Value on NNMi's Global Manager
Tenant	UUID: uniqueTenant#one Name: *MyNewestCustomer*	→ Same UUID as Regional Setting. Name: MyCustomer (name NNMi established during initial replication cycle, see 1).
Security Group	UUID: uniqueSecurityGrp#one Name: Tier1Support	UUID: uniqueSecurityGrp#two Name: MyCustomer

Newly Replicated Nodes

Node Attribute	Original Node's Attribute Value on NNMi's Regional Manager	Replicated Node's Attribute Value on NNMi's Global Manager
Tenant	UUID: uniqueTenant#one Name: *MyNewestCustomer*	→ Same UUID as Regional Setting. Name: MyCustomer (name NNMi established during initial replication cycle, see 1).
Security Group	UUID: uniqueSecurityGrp#one Name: Tier1Support	UUID: uniqueSecurityGrp#two Name: MyCustomer

3. Global Manager's NNMi administrator changes the assigned Security Group for a specific Replicated Node. Click here for details.

   (No change on the Regional Manager.)

Node Attribute	Original Node's Attribute Value on NNMi's Regional Manager	Replicated Node's Attribute Value on NNMi's Global Manager
Tenant	UUID: uniqueTenant#one Name: MyNewestCustomer	→ Same UUID as Regional Setting. Name: MyCustomer (name NNMi established during initial replication cycle, see 1).
Security Group	UUID: uniqueSecurityGrp#one Name: Tier1Support	UUID: *uniqueSecurityGrp#seven* Name: *Region1Security*

4. Regional Manager's NNMi administrator changes the assigned Security Group for a specific Node. Click here for details.

   (No change on the Global Manager.)

Node Attribute	Original Node's Attribute Value on NNMi's Regional Manager	Replicated Node's Attribute Value on NNMi's Global Manager
Tenant	UUID: uniqueTenant#one Name: MyNewestCustomer	→ Same UUID as Regional Setting. Name: MyCustomer (name NNMi established during initial replication cycle, see 1).
Security Group	UUID: *uniqueSecurityGrp#four* Name: *Building4*	UUID: uniqueSecurityGrp#seven Name: Region1Security

5. Global Manager's NNMi administrator changes the *MyCustomer* Tenant object's definition to have a different Initial Discovery Security Group: *RockyMountRegion*. Click here for details.

   Any Nodes replicated for the first time have Security Group set to the new Initial Discovery Security Group attribute value: *RockyMountRegion*.

   Newly Replicated Nodes

Node Attribute	Original Node's Attribute Value on NNMi's Regional Manager	Replicated Node's Attribute Value on NNMi's Global Manager
Tenant	UUID: uniqueTenant#one Name: MyNewestCustomer	→ Same UUID as Regional Setting. Name: MyCustomer (name NNMi established during initial replication cycle, see 1).
Security Group	UUID: uniqueSecurityGrp#four Name: Building4	UUID: *uniqueSecurityGrp#ten* Name: *RockyMountRegion*

   All previously replicated Node's Security Group settings remain unchanged (unless manually changed). NNMi does not change any Node settings when the Tenant object's Initial Discovery Security Group attribute value changes.

   Previously Replicated Node

Node Attribute	Original Node's Attribute Value on NNMi's Regional Manager	Replicated Node's Attribute Value on NNMi's Global Manager
Tenant	UUID: uniqueTenant#one Name: MyNewestCustomer	→ Same UUID as Regional Setting. Name: MyCustomer (name NNMi established during initial replication cycle, see 1).
Security Group	UUID: uniqueSecurityGrp#four Name: Building4	UUID: uniqueSecurityGrp#seven Name: Region1Security

Regional Manager: Create a Forwarding Filter (Limit the available Node information)

(*NNMi Advanced - Global Network Management feature*) As administrator of the Regional Manager, you can specify which Node object data Global Managers can access:

To provide all Node object data to Global Managers in your environment, click here.

Do nothing. NNMi automatically forwards all Node object data unless a Forwarding Filter is defined. Also see, "About Multi-Tenancy and Global Network Management" on page 95.

To limit available Node object data by creating a Forwarding Filter, click here.

(*NNMi Advanced - Global Network Management feature*) The Global Manager and the Regional Manager maintain separate sets of data. Conclusions about each Node are derived from the available data and can sometimes be different. Regional Managers forward the results of each Auto-Discovery cycle to the Global Manager. The Regional Manager can have a Node Group filter configured to limit the amount of data that is forwarded to the Global Manager. Filters are usually unnecessary for Global Network Management. Do not filter out nodes that are important for connectivity in your network environment to ensure NNMi has the data needed for accurate root cause analysis.

1. Navigate to the **Global Network Management** form:
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Select the **Global Network Management** form.
2. Select the **Forwarding Filter** tab.
3. Click the **Node Group** Lookup icon and select one of the options from the drop-down menu:
   - Show Analysis to view Analysis Pane information for the currently selected Node Group name. (See Use the Analysis Pane for more information about the Analysis Pane.)
   - Quick Find to view and select from the list of all existing Node Groups (for more information see "Use the Quick Find Window" on page 41).
   - Open to display the details of the currently configured (selected) Node Group (see Node Group form for more information).
   - New to create a new Node Group (see "Create Node Groups" on page 295 for more information).
4. Click **Save and Close**.
5. Global Managers in your network environment can now access only information about the Nodes in the specified Node Group. If any Global Managers have previously gathered a wider range of Node object data, that extra data is automatically removed from the Global Managers database.

   To verify that your Forwarding Filter is working as expected, wait until the next NNMi rediscovery cycle finishes on your NNMi management server and then log on to the Global

Manager. Follow the directions in View the NNMi Management Servers' Domain List. You should see only the members of the Node Group specified as your Forwarding Filter.

Incidents associated with the specified Nodes are not forwarded to the Global Manager. *Each server maintains an independent group of incidents.*

Regional Manager administrators can make exceptions to this for SNMP traps and NNM 6.x/7.x events. The administrator must specifically configure forwarding to the Global Managers:

- "Configure Forward to Global Manager Settings for an SNMP Trap Incident (NNMi Advanced)" on page 927
- "Configure Forward to Global Managers Settings for a Remote 6.x/7.x Event Incident (NNMi Advanced)" on page 1338

To identify these specifically forwarded SNMP traps and NNM 6.x/7.x events on the Global Manager, see Monitor Incidents in a Global Network Management Environment (NNMi Advanced).

Global Manager: Connect to a Regional Manager

(*NNMi Advanced - Global Network Management feature*) As administrator, you can set up this NNMi management server as a Global Manager that displays information from other NNMi management servers (Regional Managers).

Tip: If the group of nodes being managed by a Regional Manager includes nodes already being managed by the Global Manager, the duplicate information from the Regional Manager is not imported into the Global Manager's database. If two Regional Managers are managing the same node, only the first instance to be forwarded is added to the Global Manager's database. Also see, "About Multi-Tenancy and Global Network Management" on page 95

To enable communication from this NNMi management server to another in your network:

1. Prerequisite:

   All NNMi management servers in your network environment that participate in Global Network Management (Global Managers and Regional Managers) or Single Sign-On (SSO) must have their internal time clocks synchronized in universal time.

   > **Caution:** Use a Time Synchronization program, for example, the UNIX (HP-UX / Linux / Solaris) tool Network Time Protocol Daemon (NTPD) or one of the available Windows operating system tools.

   Review the Global Network Management deployment choices and “Configuring Single Sign-On for Global Network Management” section in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.

2. Complete the required steps described in the *HP Network Node Manager i Software Deployment Reference* (available at: `http://h20230.www2.hp.com/selfsolve/manuals`), then navigate to the **Global**

Network Management form.

a. From the workspace navigation panel, select the **Configuration** workspace.

b. Select the **Global Network Management** form.

3. Select the **Regional Manager Connections** Tab.
4. Do one of the following:
   - To create a new configuration, click the New icon.
   - To edit a configuration, click the Open icon in the row representing the configuration you want to edit.
   - DO NOT delete a configuration (the Delete icon). See "Disconnect Communication with a Regional Manager" on page 108 for more information.
5. In the **Regional Manager** form, provide the basic configuration settings (see basic settings table).
6. From the Connection tab, navigate to the **Regional Manager Connection** form (see "Global Manager: Configure the Regional Manager Connection" on the next page for more information). Do one of the following:
   - To create a new connection, click the New icon.
   - To edit a connection, select a row, click the Open icon.
   - To delete a connection configuration, select a row and click the Delete icon.
7. Click **Save and Close** to return to the Regional Manager form.
8. Click **Save and Close** to return to the Global Network Management form.
9. Click **Save and Close**. NNMi establishes communication with the specified Regional Manager. That NNMi management server now forwards information about discovery and monitoring results to this NNMi management server.

Tip: To verify that the connection is working, see "Determine the State of the Connection to a Regional Manager" on page 112.

Basic Settings for this Regional Manager (NNMi Management Server)

Attribute	Description
Name	Type a meaningful name for this configuration record about the Regional NNMi management server. For example: • The name your team uses to refer to the Regional NNMi management server. • The company site being managed by the Regional Manager. • The geographic area (Japan or Germany) being managed by the Regional Manager. The text you type appears in the Node view and NNMi Management Server view. This text string also appears in the Nodes by Management Server view's drop-down filter.

Basic Settings for this Regional Manager (NNMi Management Server), continued

Attribute	Description
	Alpha-numeric and special characters (~ ! @ # $ % ^ &amp; * ( ) _+ -) are permitted. No spaces are permitted. **Note**: Communicate this Name attribute value to your team so they understand the relationship between this name and the NNMi management server's DNS name (used to log on to that NNMi management server).
Connection State	NNMi provides the value for this attribute.
UUID	NNMi provides the value for this attribute. This is a unique number assigned by the NNMi database.
Description	*Optional*. Provide any description that would be useful for communication purposes within your team. Type a maximum of 250 characters. Alpha-numeric, spaces, and special characters (~ ! @ # $ % ^ &amp; * ( ) _+) are permitted.

Global Manager: Configure the Regional Manager Connection

(*NNMi Advanced - Global Network Management feature*) As administrator, you configure how this NNMi management server communicates with another NNMi management server in your network environment (the Regional Manager).

> **Tip:** If the group of nodes being managed by a Regional Manager includes nodes already being managed by the Global Manager, the duplicate information from the Regional Manager is not imported into the Global Manager's database. If two Regional Managers are managing the same node, only the first instance to be forwarded is added to the Global Manager's database.

To configure the communication connection to another NNMi management server:

1. Prerequisite:

   All NNMi management servers in your network environment that participate in Global Network Management (Global Managers and Regional Managers) or Single Sign-On (SSO) must have their internal time clocks synchronized in universal time.

   > **Caution:** Use a Time Synchronization program, for example, the UNIX (HP-UX / Linux / Solaris) tool Network Time Protocol Daemon (NTPD) or one of the available Windows operating system tools.

   Review the Global Network Management deployment choices and “Configuring Single Sign-On for Global Network Management” section in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.

2. Navigate to the **Regional Manager Connection** form.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Select the **Global Network Management** form
   c. Select the **Regional Manager Connections** tab.
   d. Do one of the following:
      - To create a new configuration, click the New icon.
      - To edit a configuration, click the Open icon in the row representing the configuration you want to edit.
      - DO NOT delete a configuration (the Delete icon). See "Disconnect Communication with a Regional Manager" on page 108 for more information.
   e. In the **Regional Manager** form, navigate to the Connections tab. Do one of the following:
      - To create a new connection, click the New icon.
      - To edit a connection, select a row, click the Open icon.
      - To delete a connection configuration, select a row and click the Delete icon.
3. Provide the connection configuration settings (see connection configuration settings table).

   > **Note:** If the Regional Manager participates in a high-availability (HA) environment, enter configuration settings for each server in the high-availability group (application fail-over).

4. Click **Save and Close** to return to the Regional Manager form.
5. Click **Save and Close** to return to the Global Network Management form.
6. Click **Save and Close**. NNMi establishes communication with the Regional NNMi management server. The Regional Manager forwards information about discovery and monitoring results.

> **Tip:** To verify that the connection is working, see "Determine the State of the Connection to a Regional Manager" on page 112.

Connection Configuration Settings for a Regional NNMi Management Server

Attribute	Description
Hostname	The official *fully-qualified-domain-name* of the Regional Manager (the NNMi management server). To verify the correct value, do one of the following: • Log on to the Regional Manager, select **Help** → **System Information**, and navigate to the **Server** tab. Use the value displayed in the Official Fully Qualified Domain Name (FQDN) field. • Use the nnmofficialfqdn.ovpl command.

Connection Configuration Settings for a Regional NNMi Management Server, continued

Attribute	Description
	Note: If you want NNMi to use secure sockets layer encryption (HTTPS) to access this Regional NNMi management server, the value must match the hostname as specified in that server's SSL Certificate. For information about establishing the required trust relationship, see the"Global Network Management" chapter in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals.` NNMi uses this hostname for communication with the Regional NNMi management server and to construct URL Actions. See "Authentication Requirements for URLs Access" on page 1491. See also "Actions Provided by NNMi" on page 43 and read about these actions: • **Actions** → **Regional Manager Console** (opens the NNMi console) • **Actions** → **Open from Regional Manager** (opens the Node form)
Use Encryption	If □ disabled, NNMi uses hypertext transfer protocol (HTTP) and plain sockets to access this Regional NNMi management server. If ☑ enabled, NNMi uses secure sockets layer encryption (HTTPS/SSL) to access this Regional NNMi management server.
HTTP(S) Port	The Global Manager initiates all communication sockets. The Global Manager needs access to the following default TCP ports on each Regional Manager: **Non-Encrypted** • nmsas.server.port.web.http = 80 • nmsas.server.port.hq = 4457 **Encrypted** • nmsas.server.port.web.https = 443 • nmsas.server.port.hq.ssl = 4459 To determine the current port number configuration or change port settings, access the Regional Manager and look in the nms-local.properties file. See the nnm.ports Reference Page for more information. If □ Use Encryption is disabled (previous attribute), enter the port number for HTTP access to the NNMi console on the Regional NNMi management server. For example `http://<serverName>:<portNumber>/nnm/` If ☑ Use Encryption is enabled (previous attribute), enter the port number for HTTPS access to the NNMi console on the Regional NNMi management server. For example `https://<serverName>:<portNumber>/nnm/`
User	Type the user name required for NNMi sign-in for the **system** account on this

Connection Configuration Settings for a Regional NNMi Management Server, continued

Attribute	Description
Name	Regional NNMi management server.
User Password	Type the password for the NNMi **system** account on this Regional NNMi management server. **Note:** NNMi encrypts the password and displays asterisks for this attribute. If you want to change the password, first clear the asterisks displayed in the Password attribute and enter the new Password value.
Ordering	A numeric value. NNMi checks for configuration settings in the order you define (lowest number first). NNMi uses the first match found for each address. Provide a unique connection ordering number for each Regional Manager configuration. Any duplicate Ordering numbers are checked in random order, for example that group of Regional Manager Connections can be checked in any order during each discovery cycle. **Tip:** Consider incrementing Ordering numbers by 10s or 100s to provide flexibility over time.

Disconnect Communication with a Regional Manager

(*NNMi Advanced - Global Network Management feature*) As administrator, you can disconnect communication between a Global Manager (NNMi management server) and a Regional Manager (another NNMi management server within your network environment).

To disconnect communication with a Regional Manager:

1. On the Global Manager (NNMi management server), navigate to the **Global Network Management** form:
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Select the **Global Network Management** form.
2. Select the **Regional Manager Connections** tab.
3. Click the Open icon in the row representing the configuration you want to edit.

   In the **Regional Manager** form, delete all Connection objects:
   a. Select the **Connections** tab.
   b. Select all Connection records, and click the ✖ Delete icon.

4. Click **Save and Close** to return to the Global Network Management form. NNMi disables communication from this Global Manager (NNMi management server) to that Regional Manager (NNMi management server).

5. In the **Regional Manager Connections** tab, note the **Name** attribute value for that connection configuration (case-sensitive). You need to type this text string to replace <*RegionalNNMiServerName*> in a later step.

6. Click **Save and Close**.

7. On the Global Manager (NNMi management server), at the command line, type the following command (see "Delete Nodes" on page 1602 and nnmnodedelete.ovpl for more information):

   > **Note:** The original *node records* on the Regional Manager (NNMi management server) are not affected. Only the *copy of the node records* will be deleted from the Global Manager's database.
   >
   > If you do not want to enter an NNMi User Name attribute value and an NNMi Password attribute value at the command line, you can use the nnmsetcmduserpw.ovpl command to specify the valid user name and password (instead of `-u` and `-p`). The credentials set using the `nnmsetcmduserpw.ovpl` command are valid for command execution by the same user. See "Set Up Command Line Access to NNMi" on page 577 for more information.

   Windows:
   ```
   %NnmInstallDir%\bin\nnmnodedelete -rm <RegionalNNMiServerName> -u
   <NNMiadminUserName> -p <NNMiadminPassword>
   ```

   UNIX:
   ```
   /opt/OV/bin/nnmnodedelete -rm <RegionalNNMiServerName> -u
   <NNMiadminUserName> -p <NNMiadminPassword>
   ```

   NNMi searches the Global Manager's database for all nodes that this Regional Manager is responsible for monitoring in your network environment. NNMi removes the node records from the Global Manager's database (these node records represent information *forwarded from* the Regional Manager). NNMi removes all associated data:

   - Any interface or IP address information belonging to a deleted node.
   - Any discovery seeds that match the name or IP address of a deleted node (unless you use the `nmnodedelete -keepSeed` option).

   Each Incident associated with the deleted Node is modified in the following ways, but not deleted from the NNMi database: The **Status** attribute changes to **Closed**. The **Correlation Notes** indicate the deletion of the associated node, interface, or address. The **RCA State** attribute changes to **FALSE**. Incidents generated from SNMP traps or NNM 6.x/7.x Events (received from the deleted Node) appear in the Incident views, but remain unresolved.

   To remove the Incidents from your NNMi database, follow the instructions in "Archive and Delete Incidents" on page 1598 to delete "Closed" Incidents. You will be deleting all "Closed" Incidents, not just the "Closed" Incidents associated with this Regional Manager.

8. On the Global Manager (NNMi management server), remove the configuration record for this Regional Manager.

a. From the workspace navigation panel, select the **Configuration** workspace.

b. Select the **Global Network Management** form.

c. Select the **Regional Manager Connections** tab.

d. Select the row that represents the Regional Manager (NNMi management server) that should no longer communicate with this NNMi management server (the Global Manager), and click the ✖ Delete icon.

e. Click 💾 **Save and Close**.

9. NNMi no longer requests information about discovery and monitoring results from that Regional Manager.

> **Note:** The NNMi management server that is no longer one of the Regional Managers is still fully-functioning, but communication between the two NNMi management servers is now disabled.
>
> Traps from that Regional Manager are still forwarded to the Global Manager if configured to do so, see "Configure Trap Forwarding Destinations" on page 1381. Disable any trap forwarding that you no longer need.

Troubleshoot Global Network Management

(*NNMi Advanced - Global Network Management feature*) The Global Manager and the Regional Manager maintain separate sets of data. Conclusions about each Node are derived from the available data and can sometimes be different. Regional Managers forward the results of each Spiral Discovery cycle to the Global Manager. The Regional Manager can have a Node Group filter configured to limit the amount of data that is forwarded to the Global Manager. Filters are usually unnecessary for Global Network Management. Do not filter out nodes that are important for connectivity in your network environment to ensure NNMi has the data needed for accurate root cause analysis.

- The Global Manager might know information about why a connection from one site to another is down, but the Regional Manager just knows that the router connected to that remote site has an interface that is down. Use **Actions** → **Regional Manager Console** to see the other perspective.
- When troubleshooting a Node on the Global Manager, you can use **Actions** → **Open from Regional Manager** to see the latest Node information on the Regional Manager.

(*NNMi Advanced - Global Network Management feature*) This group of help topics can help you troubleshoot any problems with Global Network Management:

- "Clock Synchronization Issues (SSO / Global Network Management)" on the next page
- "Determine the State of the Connection to a Regional Manager" on page 112
- "Check the Health of Global Managers and Regional Managers" on page 113
- "Node Synchronization Issues " on page 115

Watch for these Incidents (error messages):

- "Error Messages About Regional Managers (NNMi Advanced)" on page 117
- Message Queue Size Exceeded Limit
- Forwarded Incident Rate Exceeded
- Queue Size Exceeded Limit

If you suspect problems, see the following NNMi log file on each NNMi management server for details about any communication problems between the Global Manager and Regional Manager:

- **Windows:**
  `%NnmDataDir%\log\nnm\nnm.0.0.log`
- **UNIX:**
  `/var/opt/OV/log/nnm/nnm.0.0.log`

See also these topics in NNMi Help for Operators:

- Is the Global Network Management Feature Enabled?
- View the NNMi Management Servers' Domain List

Clock Synchronization Issues (SSO / Global Network Management)

(Single Sign-On and *NNMi Advanced - Global Network Management feature*)

All NNMi management servers in your network environment that participate in Global Network Management (Global Managers and Regional Managers) or Single Sign-On (SSO) must have their internal time clocks synchronized in universal time.

> **Caution:** Use a Time Synchronization program, for example, the UNIX (HP-UX / Linux / Solaris) tool Network Time Protocol Daemon (NTPD) or one of the available Windows operating system tools.

Review the Global Network Management deployment choices and “Configuring Single Sign-On for Global Network Management” section in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.

- For clock issues when creating Regional Manager Connections, click here.

  If you see the following message at the bottom of the NNMi console:

  ```
  NNMi is not connected to 1 Regional Manager. See Help → System
  Information, Global Network Management.
  ```

  Check the `nnm.0.0.log` file on the Global Manager for the following message:

  ```
  WARNING: Not connecting to system <serverName> due to clock
  difference of <number of seconds>. Remote time is <date/time>.
  ```

- If Regional Manager Connections break after running successfully, click here.

  Perhaps the clocks are no longer synchronized. Check the `nnm.0.0.log` file on the Global Manager for the following message:

```
WARNING: Not connecting to system <serverName> due to clock
difference of <number of seconds>. Remote time is <date/time>.
```

Within a few minutes of this warning, NNMi disconnects the Regional Manager Connection. And the following message appears at the bottom of the NNMi console:

```
NNMi is not connected to 1 Regional Manager. See Help → System
Information, Global Network Management.
```

Determine the State of the Connection to a Regional Manager

(*NNMi Advanced - Global Network Management feature*) NNMi provides the **Connection State** attribute to help you track the health of communication connections between Global Managers and Regional Managers in your network environment. The table below describes each possible Connection State value.

To verify the state of the communication connection between NNMi management servers:

1. Open the NNMi console on the Global Manager (NNMi management server).
2. Navigate to the **Global Network Management** form.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Select the **Global Network Management** form.
3. Select the **Regional Manager Connections** tab.
4. Locate the **Connection State** column in this view.
5. Check the Connection State value for each Regional NNMi management server.

   Tip: To verify the list of Nodes being managed by each NNMi management server, see View the NNMi Management Servers' Domain List.

Possible States for Regional Manager Connections

Connection State	Description
Not Established	The connection configuration was recently saved, and NNMi is attempting to establish the connection.
Partial Connection	The connection state is transitioning between states due to a recent change in your network environment or a change in NNMi configuration settings.
Connected	Communication between the two NNMi management servers is working properly.
Not Connected	An error occurred and the connection failed. Check the Regional Management Server configuration settings. Perhaps one of the designated port numbers is not correct? See "Global Manager: Connect to a Regional Manager" on page 103. Perhaps the Regional NNMi management server is currently down? See "Troubleshoot Global Network Management" on page 110.

Thresholds in the Global Network Management Environment

(*NNMi Advanced*) When using the NNMi Global Network Management feature: Configure thresholds carefully as follows:

- Monitoring Configuration: Interface Group and Node Group thresholds are configured on each NNMi management server (Regional or Global) that is responsible for the objects being monitored (Interface, Node Component, Node). The threshold results are automatically communicated from Regional Managers to Global Managers (but not visa versa):
  - "Configure Threshold Monitoring for Interface Groups" on page 381
  - "Configure Threshold Monitoring for Node Groups" on page 402
- Custom Poller Collection thresholds are configured on the NNMi management server (Regional or Global) that is responsible for the objects being monitored. The results are *not* communicated to other NNMi management servers.
  - "Configure Threshold Information for a Custom Poller Collection" on page 441

  > **Tip:** Although Custom Polled Instances are not sent from a Regional Manager (NNMi management server) to the Global Manager. From the Global Manager, users can access that information by opening the monitored object's form and clicking **Actions** → **Open from Regional Manager** to see the list of Custom Polled Instances on the Regional Manager.
- Trap volume/forwarding is configured on each NNMi management server (Regional or Global).
  - Interpret Incidents Related to SNMP Traps

Check the Health of Global Managers and Regional Managers

Do one of the following to check the health of the Global Network Management feature:

- Log on to the Global Manager as an NNMi administrator, and open the NNMi console on the Global Manager (NNMi management server). Click here for more information.
  a. Click the **Help** → **System Information**.
  b. Click the **Global Network Management** tab.
  c. In the **Regional Managers Reporting to this Global Manager** section, review the list of all Regional Managers that report to this Global Manager:
     - **Name**: The current value of the Name attribute for this Regional NNMi management server (as specified in the Remote Manager Connection form).
     - **Connection State**: The current state of communication between the Global Manager and Regional Manager. There are four possible values:
       - **Not Established** — A new Regional Manager Connection is not yet fully functional. This state is brief unless NNMi encounters a problem.

        - **Connected** — Data is flowing between the Global Manager and the Remote Manager.
        - **Not connected** — A previously established connection is no longer working. See "Clock Synchronization Issues (SSO / Global Network Management)" on page 111.

          All NNMi management servers in your network environment that participate in Global Network Management (Global Managers and Regional Managers) or Single Sign-On (SSO) must have their internal time clocks synchronized in universal time.

          > **Caution:** Use a Time Synchronization program, for example, the UNIX (HP-UX / Linux / Solaris) tool Network Time Protocol Daemon (NTPD) or one of the available Windows operating system tools.

          Review the Global Network Management deployment choices and “Configuring Single Sign-On for Global Network Management” section in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.
    - **Node Count**: The number of nodes in the Global Manager's database that are being managed by this Regional Manager.

- Log on to the Regional Manager as an NNMi administrator, and open the NNMi console on the Regional Manager (NNMi management server).click here for more information..
  a. Click the **Help** → **System Information**.
  b. Click the **Global Network Management** tab.
  c. Scroll down to the **Reporting to Global Managers** section, and review the list of all Global Managers that receive data from this Regional Manager:
    - **Name**: The fully-qualified DNS hostname of the Global Manager (NNMi management server).

      Note: If you see something other than a fully-qualified DNS hostname in the Name column, the Global Manager is down and has been down since this Regional Manager was last restarted (see "Stop or Start an NNMi Process" on page 82 or "Stop or Start NNMi Services" on page 86 for more information).
    - **Messages Currently in Queue**: The current number of messages that need to be sent to the Global Manager.

      Messages are automatically sent to the Global Manager. If the number of messages in the queue continually increases and never decreases, or if the number of messages in the queue consistently exceeds 10,000, then there might be a problem.

      Note: If the Global Manager is down for maintenance for a few hours, the queue size naturally increases until the Global Manager is back online.

      Queue size over 100,000 indicates a serious issue. Consider disconnecting that global manager until the issue can be resolved..
- NNMi administrators can use the command line on any NNMi management server to generate a report about NNMi health. See the nnmhealth.ovpl Reference Page for more information.

There are two ways to log on to a Regional Manager:

- Directly log on to the Regional Manager (NNMi management server).
- From the Global Manager, select any Node being managed by the Regional Manager and click **Actions → Regional Manager Console**. See "Actions Provided by NNMi" on page 43.

Node Synchronization Issues

(*NNMi Advanced - Global Network Management feature*).

Note: The Global Manager and the Regional Manager maintain separate sets of data. Nodes that are managed by the Regional Manager are discovered on the Regional Manager and are not rediscovered by the Global Manager.

Use the nnmnoderediscover.ovpl command when information about one or more nodes on the Global Manager or on a Regional Manager is not as expected or up-to-date. This is an unlikely scenario, but could be caused by data loss resulting from disk corruption, operator error, or extended downtime of the Global Manager. This command enables you to request that the specified Regional Manager send the most recent discovery information to the Global Manager. You can choose to send information for all nodes or for a subset of nodes.

Tip: Begin by re-synchronizing the smallest set of nodes that appear to have inconsistencies. If you need to re-synchronize all nodes in your managed network, execute this command during off hours when possible.

The nnmnoderediscover.ovpl command places the node or nodes into the NNMi discovery queue. The amount of time before the node starts discovery depends on how long NNMi takes to work through the nodes in the queue.

Caution: Use nnmnoderediscover.ovpl and especially `nnmnoderediscover.ovpl -fullsync` with care. Rediscovering all nodes or a large subset of nodes causes a large increase in CPU usage and network bandwidth. The `-fullsync` option with a large number of nodes also can cause a large increase in resource usage due to the increase in status recalculations.

You can re-synchronize discovery information for any of the following:

- All the nodes in your network (from the Global Manager) or a subset of nodes that are handled by a Regional Manager
- All the nodes managed by the local NNMi management server
- All of the nodes listed in a specified file or a single node

For example, to re-synchronize discovery information for all nodes on a specified Regional Manager, from the Global Network manager, use the following syntax:

```
nnmnoderediscover.ovpl -rm <regional_manager>
```

When you want to force the re-synchronization of all information about all nodes managed by an NNMi Regional Manager, including State and Status information, use the `-fullsync` option as shown in the following example:

```
nnmnoderediscover.ovpl -rm <regional_manager> -fullsync
```

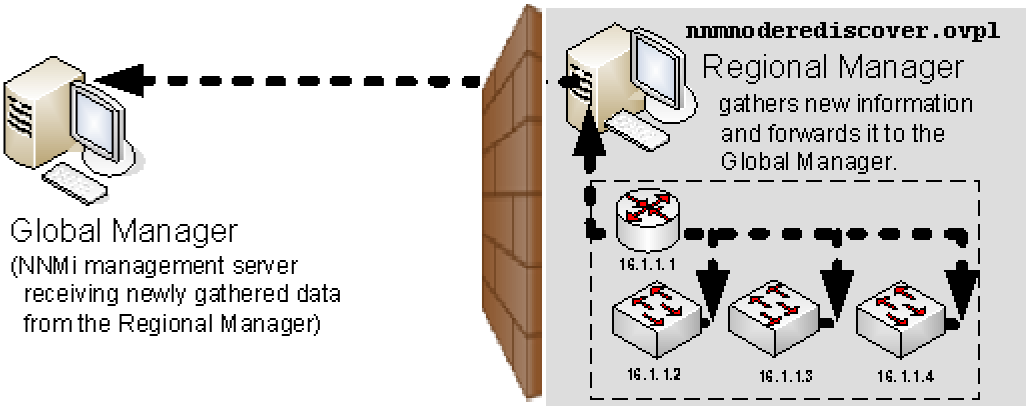
NNMi automatically performs a full re-synchronization in the following cases:

- When upgrading an NNMi management server from an earlier NNMi release
- After restoring an NNMi management server from a backup.
- After failover in an NNMi cluster. For more information about NNMi’s Application Failover feature, see in the "Resilience" chapter of the *HP Network Node Manager i Software Deployment Reference* which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.

When using nnmnoderediscover.ovpl -fullsync, note the following:

- When NNMi synchronizes information for locally managed nodes, NNMi does the following:
  - Performs a Configuration Poll (nnmconfigpoll.ovpl) for each node specified.
  - Reloads and refreshes the monitoring configuration for the node
  - State Poller sends all current State values to the Causal Engine for analysis.
  - The Causal Engine recalculates the Status for each node specified using the current State information.
  - If the NNMi management server is a Regional Manager, the re-synchronized information is automatically uploaded to the Global Manager.

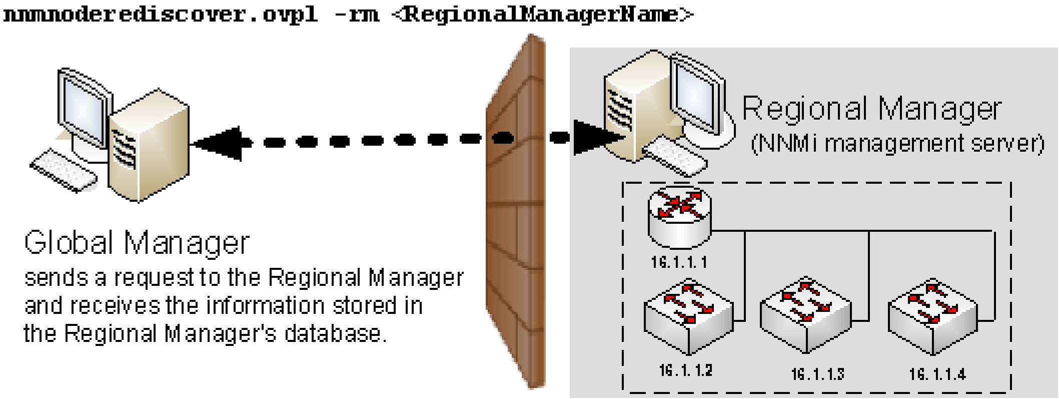
  The following diagram illustrates executing `nnmnoderediscover.ovpl -fullsync` locally on the Regional Manager.


nnmnoderediscover.ovpl
Regional Manager
gathers new information and forwards it to the Global Manager.
16.1.1.1
16.1.1.2
16.1.1.3
16.1.1.4
Global Manager
(NNMi management server receiving newly gathered data from the Regional Manager)

  When NNMi synchronizes information for remotely managed nodes (for example using a `nnmnoderediscover.ovpl -rm` command from the Global Manager), NNMi does not execute an `nnmconfigpoll.ovpl` command for each node. Instead, the Global Manager requests the Node information that is currently stored in the Regional Manager's database.

  The following diagram illustrates executing the following command on the Global Manager:

```
nnmnoderediscover.ovpl -rm <regional_manager>
```



```
nnmnoderediscover.ovpl -rm <RegionalManagerName>
```

Global Manager sends a request to the Regional Manager and receives the information stored in the Regional Manager's database.

See nnmnoderediscover.ovpl, nnmconfigpoll.ovpl and nnmstatuspoll.ovpl for more information.

Error Messages About Regional Managers (*NNMi Advanced*)

(*NNMi Advanced - Global Network Management feature*) A special set of incidents keeps the Global Manager informed of any problems with the Regional Manager:

- Licensing issues
  - License Expired
  - License Mismatch
  - License Node Count Exceeded
- Application fail-over health issues
  - Nnm Cluster Failover
  - Nnm Cluster Lost Standby
  - Nnm Cluster Startup
  - Nnm Cluster Transfer
- Traffic volume issues
  - Snmp Trap Limit Critical
  - Snmp Trap Limit Major
  - Snmp Trap Limit Warning
  - Trap Storm

These incidents are generated on the Regional Manager (NNMi management server). The Regional Manager forwards a copy of these incidents to the Global Manager. NNMi dynamically closes these incidents on the Regional Manager when the issue is resolved. The NNMi administrator for the Global Manager (NNMi management server) must manually close the forwarded copy.

From any Incident view, you can identify the forwarding server or servers (`cia.remotemgr`). Use the Custom Incident Attribute tab on the Incident form for the selected incident. NNMi uses Custom Incident Attributes (CIAs) to attach additional information to incidents.

Chapter 6

Configuring Communication Protocol

NNMi uses the following protocols to discover your network and monitor the health of your network environment:

- Simple Network Management Protocol (SNMPv1 and SNMPv2c)
  - Read-only queries, also known as "Get" commands.

    SNMPv1 and SNMPv2c require the use of a read community string to authenticate messages that are sent between NNMi and SNMP agents. NNMi cannot discover information about the SNMPv1 and SNMPv2c devices in your network environment until you provide the appropriate read community strings. During discovery and monitoring, NNMi uses the read community strings you provide in the Communication Configurations option of the 🔧 Configuration workspace. When a device is first discovered, NNMi tries all appropriate read community strings and makes a record of the first read community string that works. To keep network traffic to a minimum, from then on NNMi uses the recorded read community string when communicating with that device using SNMP. If at some point the device no longer responds to the recorded read community string, NNMi tries all appropriate read community strings and makes a record of the first read community string that now works.
  - Write commands, also known as "Set" commands.

    SNMPv1 and SNMPv2c require the use of a write community string to authenticate messages that are sent between the nnmsnmpset.ovpl command and SNMP agents.
- SNMPv3 requires the use of user-based security model (USM) user names instead of *SNMPv1/SNMPv2c community strings* to authenticate messages that are sent between NNMi and SNMP agents. NNMi cannot discover information about the SNMPv3 devices in your network environment until you provide the appropriate user name and authentication. During discovery and monitoring, NNMi uses the *SNMPv3 User Name* attribute value and authentication that the NNMi administrator provides in the Communication Configuration workspace. When a device is first discovered, NNMi tries all appropriate USM user names and makes a record of the first USM user name that works. To keep network traffic to a minimum, from then on NNMi uses the recorded *SNMPv3 User Name* attribute value when communicating with that device using SNMP. If at some point the device no longer responds to the recorded *SNMPv3 User Name* attribute value, NNMi tries all appropriate USM user names and makes a record of the one that now works.
- Internet Control Message Protocol (ICMP) ping commands

Note: If NNMi discovers a device for which no SNMP authentication was provided in the Communication Configuration workspace, that device is treated as a non-SNMP device.

You control the amount of traffic NNMi generates on your network. You can modify the settings to meet your needs.

To configure the way NNMi uses ICMP and SNMP protocols, do the following:

1. Navigate to the **Communication Configuration** form.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Select the **Communication Configuration**.
2. Make your configuration choices. The Communication Configuration settings determine whether each NNMi Rediscovery cycle automatically detects the best SNMP choice (v1, v2, or v3) for each Node (automatically detects any upgrade to the SNMP agent on each Node), or uses only the SNMP version that you specify.

   Click here for a list of choices .
3. Click **Save and Close** to apply your changes.

   Note: You control the amount of network traffic generated by NNMi by designating the **Rediscovery Interval** setting (see "Adjust the Rediscovery Interval" on page 210 for more information) and making choices when you "Configure NNMi Monitoring Behavior" on page 340.

Configure Default SNMP, Management Address, and ICMP Settings

NNMi generates network traffic using ICMP and SNMP protocols to discover and monitor your network environment. Default settings for the use of these protocols are provided, for example timeout and retry behavior settings.

NNMi Advanced: The NNMi administrator specifies whether your network environment uses IPv4 or IPv6 addresses for SNMP agents Management Address. See Management Address Selection table.

To configure the default communication protocol settings for your environment:

1. Navigate to the **Communication Configuration** form.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Select the **Communication Configuration**.
2. Locate the **Default Settings** groups.
3. Make your configuration choices (see the Default SNMP Settings table, Management Address Selection table, and Default ICMP Settings table).

   For an explanation of how NNMi implements timeout and retry configurations, see "Timeout / Retry Behavior Example for SNMP" on page 127 and "Timeout / Retry Behavior Example for ICMP" on page 128.
4. Click **Save and Close** to apply your changes.

Default SNMP Settings Attributes

Attribute	Description
Enable SNMP	**Note:** The NNMi administrator can over-ride this setting for a Region or on a

Default SNMP Settings Attributes, continued

Attribute	Description
Address Rediscovery	per-node basis. See "Communication Region Form" on page 137and "Specific Node Settings Form (Communication Settings)" on page 155. If ☑ enabled, NNMi automatically identifies which management address (SNMP agent) to use for each device. If the initially configured address becomes unreachable, NNMi automatically locates another address, if possible, and changes the management address attribute value. Click here for more information. When NNMi first discovers a node, the *seed address* (provided by the NNMi administrator) or discovered address (for non-seeded nodes) becomes the initial address used for SNMP communication. After NNMi builds an inventory of all IP addresses associated with the node (see "How Spiral Discovery Works" on page 176), NNMi follows a set of rules to determine which address is the best choice for each node's Management Address: **Note:** With NNMi Advanced, the NNMi administrator specifies whether NNMi prefers IPv4 or IPv6 addresses when selecting the Management Address. See Configure Default SNMP, Management Address, and ICMP Settings. 1. NNMi ignores the following addresses when determining which Management Address is most appropriate: ▪ Any address of an administratively-down interface. ▪ Any address that is virtual (for example, **VRRP**[1]). ▪ Any IPv4 **Anycast Rendezvous Point IP Address**[2] or IPv6 Anycast address. ▪ Any address in the reserved loopback network range. IPv4 uses 127/24 (`127.*.*.*`) and IPv6 uses `::1`. ▪ Any **IPv6 link-local address**[3]. 2. If the NNMi Administrator specifies Enable SNMP Address Rediscovery in Communication Configuration, NNMi prefers the last-known Management Address (if any). 3. If the Management Address does not respond and the NNMi Administrator

[1] Virtual Router Redundancy Protocol

[2] Rendezvous Point addresses are loopback addresses used for routers in multi-cast network configurations.

[3] A non-routable IPv6 unicast address only used for communication with other nodes on the same link (LAN or VLAN). Link local addresses cannot be used for communication that must be forwarded through a router. IPv6 auto-configuration automatically assigns a unique link local address in the fe80::/10 address space to each IPv6-enabled interface on a system.

Default SNMP Settings Attributes, continued

Attribute	Description
	specifies Enable SNMP Address Rediscovery in Communication Configuration, NNMi uses the Communication Configuration settings for *Management Address Selection*. The NNMi Administrators chooses the order in which NNMi checks the following: ▪ Seed IP / Management IP - If the NNMi Administrator configures a Seed, NNMi uses the Seed address (either a specified IP address or the DNS address associated with a specified hostname) only during initial Discovery. NNMi then requests the current Management Address (the address from which the node's SNMP Agent responds) and uses that IP address for all communication after initial discovery. ▪ Lowest Loopback - If a node supports multiple **loopback address**[1], NNMi queries each loopback addresses, starting with the lowest number. NNMi uses the loopback address with the lowest number from which the SNMP agent responds (for example, 10.16.42.197 is a lower number than 10.16.197.42). ▪ Highest Loopback - If a node supports multiple **loopback address**[2], NNMi queries each loopback addresses, starting with the highest number. NNMi uses the loopback address with the highest number from which the SNMP agent responds. ▪ Interface Matching - The NNMi Administrator chooses which interface MIB variable NNMi queries to detect changes. NNMi can use the following MIB-II attribute values: `ifIndex`, `ifName`, `ifDescr`, `ifAlias`, or a combination of these (`ifName` or `ifDescr`, `ifName` or `ifDescr` or `ifAlias`). 4. If no response, NNMi queries any remaining IP addresses in the node's IP address inventory, starting with the lowest number. NNMi uses the address with the lowest number from which the SNMP agent responds. 5. If no response, NNMi checks for any Mapped Address configured for one of the currently known addresses (see the Mapped Address column in the Custom IP Addresses view). **Note:** The address represents a *static* Network Address Translation

[1] The address associated with the loopback interface. The loopback interface is a virtual interface on a device that provides a route for internal communication. Many vendors provide a specially configured loopback for management purposes. Exact details of how loopbacks are configured varies by vendor and model. See each device's documentation for details. NNMi identifies these loopback addresses by using ifType Number 24, softwareloopback from the IANA ifType-MIB.

[2] The address associated with the loopback interface. The loopback interface is a virtual interface on a device that provides a route for internal communication. Many vendors provide a specially configured loopback for management purposes. Exact details of how loopbacks are configured varies by vendor and model. See each device's documentation for details. NNMi identifies these loopback addresses by using ifType Number 24, softwareloopback from the IANA ifType-MIB.

Default SNMP Settings Attributes, continued

Attribute	Description
	(NAT) pair's *external IP address* from the internal/external IP address pair. NNMi Administrators configure these pairs using the Overlapping IP Address Mapping form. NNMi uses this list of addresses starting with IPv4 from low to high, then IPv6 from low to high. 6. If no response, NNMi might be configured to repeat the sequence using SNMPv1, SNMPv2c, or SNMPv3 in the order specified by the NNMi administrator (Communication Configurations *SNMP Minimum Security Level* settings). 7. When all else fails, NNMi retains the last known Management Address (if any) and automatically changes the State of that SNMP Agent object to Critical. This process is repeated during each Spiral Discovery cycle, and the Management Address can change. For example, NNMi's inventory of addresses for the node expands, or the current Management Address does not respond to SNMP queries due to network problems or node reconfiguration. The NNMi administrator can prevent changes to the management address using the Communication Configurations *Enable SNMP Address Rediscovery* or *Preferred Management Address* setting. If ☐ disabled, when the current management address (SNMP agent) becomes unreachable, NNMi reclassifies the node as a non-SNMP node until the previously configured management address is available again.
Enable SNMP GetBulk	*Applies only to SNMPv2 or higher.* If you have devices in your network environment that have trouble responding to `GetBulk` commands, you can instruct NNMi to use `Get` or `GetNext` instead of `GetBulk`. If ☑ enabled, NNMi uses the SNMPv2c `GetBulk` command to gather information from devices in your network environment. If ☐ disabled, NNMi uses the SNMP `Get` or `GetNext` command to gather information from devices in your network environment (requesting responses for one SNMP OID at a time).
SNMP Timeout	(Seconds:Milliseconds) Maximum 1 millisecond less than a minute: 59 seconds 999 milliseconds. Time that NNMi waits for a response to an SNMP query before reissuing the request. Both the Discovery Process and the State Poller Service use this setting. For an explanation of how NNMi implements timeout and retry configurations, see "Timeout / Retry Behavior Example for SNMP" on page 127.
SNMP Retries Count	Maximum number of retries that NNMi issues for an SNMP query before determining the query result to be "unresponsive". Zero means no retries. Both the Discovery Process and the State Poller Service use this setting.
SNMP Port	Default is 161. Specifies the NNMi management server's port that NNMi uses

Default SNMP Settings Attributes, continued

Attribute	Description
	when generating SNMP traffic. Both the Discovery Process and the State Poller Service use this setting.
SNMP Proxy Address	*Optional*. IP address of the your SNMP Proxy Server (for example, a proxy that gathers data from non-SNMP devices and can use that data to respond to NNMi SNMP requests). To enable a proxy, you must also provide the port number of your SNMP Proxy Server. See SNMP Proxy Port (next attribute). **Note:** When you configure NNMi to use a Proxy Server, you must ensure that the Proxy Server vendor supports the Object Identifiers used to handle SNMP requests and responses. See the “NNMi Communications” chapter of the *HP Network Node Manager i Software Deployment Reference* for more information.
SNMP Proxy Port	*Optional*. Port number of the SNMP Proxy Server. To enable a proxy, you must also provide the IP address of your SNMP Proxy Server. See SNMP Proxy Address (previous attribute). **Note:** When you configure NNMi to use a Proxy Server, you must ensure that the Proxy Server vendor supports the Object Identifiers used to handle SNMP requests and responses. See the “NNMi Communications” chapter of the *HP Network Node Manager i Software Deployment Reference* for more information.
SNMP Minimum Security Level	This setting determines whether each NNMi Rediscovery cycle automatically detects the best SNMP choice (v1, v2, or v3) for each Node (automatically detects any upgrade to the SNMP agent on each Node), or uses only the SNMP version that you specify. For SNMPv1 or SNMPv2c, configure NNMi to use Community Strings in your network environment: • Community Only (SNMPv1) NNMi tries only SNMPv1 settings. • Community Only (SNMPv1 or v2c) NNMi first tries to use SNMPv2c settings, and, if that fails, NNMi tries SNMPv1 settings. • Community NNMi first tries to use SNMPv2c settings, and, if that fails, NNMi tries SNMPv1 settings. If both SNMPv2c and SNMPv1 fail, NNMi tries SNMPv3 settings if any are available. For SNMPv3, configure NNMi to use the User-based Security Module (USM) level of security required in your network environment (if your environment also

Default SNMP Settings Attributes, continued

Attribute	Description
	uses SNMPv1/SNMPv2c, select Community): • No Authentication, No Privacy • Authentication, No Privacy • Authentication, Privacy See"Timeout / Retry Behavior Example for SNMP" on page 127 for an explanation of NNMi behavior with each of these choices.

> **Note:** NNMi needs to know which SNMPv1 or SNMPv2c community strings (read/write) are used in your environment (see "Configure Default Community Strings (SNMPv1 or SNMPv2c) " on page 129) and which SNMPv3 USM settings are used in your environment (see "Configure Default SNMPv3 Settings" on page 133).

Management Address Selection Settings

Attribute	Description
First Choice	Configure how NNMi chooses the Management Address for Nodes, if possible: • Seed IP / Management IP NNMi uses the Seed address only during initial Discovery. The Seed address is either the specified IP address or the DNS address associated with the specified hostname. See "Specify Discovery Seeds" on page 256 for more information. Otherwise, NNMi uses the current Management Address. • Lowest Loopback IP address (**loopback address**[1]) • Highest Loopback IP address • Interface Matching (instead of addresses)
Second Choice	Configure how NNMi choose the Management Address for Nodes when the First Choice is not available.
Third Choice	Configure how NNMi choose the Management Address for Nodes when the First Choice and Second Choice are not available.
Interface Matching	*Optional*. When First, Second, or Third Choice is set to **Interface Matching**, provide the appropriate values for the following SNMP MIB-II attributes.

[1] The address associated with the loopback interface. The loopback interface is a virtual interface on a device that provides a route for internal communication. Many vendors provide a specially configured loopback for management purposes. Exact details of how loopbacks are configured varies by vendor and model. See each device's documentation for details. NNMi identifies these loopback addresses by using ifType Number 24, softwareloopback from the IANA ifType-MIB.

Management Address Selection Settings, continued

Attribute	Description
	Provide more than one value by separating each with a comma. Space characters are permitted within values, but not before or after a comma. For example, `Lo0,My Favorite Interface,Lo1` produces the following results with the spaces "My Favorite Interface" However, `Lo0, My Favorite Interface, Lo1` produces the following results with spaces " My Favorite Interface" (initial character is a space) and " Lo1" (initial character is a space) No wildcards or quotes allowed within values: • `ifIndex` values (for example, 4) • `ifAlias` values (for example, Vlan99) • `ifName` values (for example, lo0) • `ifDescr` values (for example, 1000Gbic Port 9/27) NNMi searches current interface data for an exact match in this order: index, alias, name, and description.
IP Version Preference	**Tip:** This attribute does not appear under Management Address Selection Settings until the NNMi Administrator follows the instructions for enabling IPv6 in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`. **IP Version Preference**: Select one of the following to influence Spiral Discovery's evaluation of *newly discovered nodes*. Previously established Management Addresses will not change if you modify this IP Version Preference setting: • IPv4 • IPv6 • Any (either IPv4 or IPv6) **Tip:** When set to Any, Spiral Discovery gives preference to IPv4 addresses when determining the Management Address of *newly discovered nodes*.

Default ICMP Settings

Attribute	Description
ICMP Timeout	(Seconds:Milliseconds) Maximum 1 millisecond less than a minute: 59 seconds 999 milliseconds. Time that NNMi waits for a response to an ICMP query before reissuing the request. For an explanation of how NNMi implements timeout and retry configurations, see

Default ICMP Settings, continued

Attribute	Description
	"Timeout / Retry Behavior Example for ICMP" on the next page.
ICMP Retries Count	Maximum number of retries that NNMi issues for an ICMP query before logging an error. Zero means no retries.

Related Topics:

"Configure Default Community Strings (SNMPv1 or SNMPv2c)" on page 129

"Configure Regions (Communication Settings)" on page 136

"Configure Specific Nodes (Communication Settings)" on page 154.

Timeout / Retry Behavior Example for SNMP

When NNMi attempts to contact a device, your configuration settings for Timeout and Retry influence NNMi behavior.

NNMi attempts to obtain information about a hostname/IP-address using SNMP, then waits the configured timeout interval for a response. If not successful, NNMi increments the timeout interval before trying again. This process repeats until one of the following is true:

- The device responds to SNMP.
- The maximum configured number of SNMP Retries fails. For example, if your timeout is 2 seconds and your retry is 3:
  - NNMi attempts to communicate with a device and waits 2 seconds for a response.
  - If unsuccessful, NNMi retries and waits 4 seconds for a response.
  - If unsuccessful, NNMi retries a second time and waits 6 seconds for a response.
  - If unsuccessful, NNMi retries a third time and waits 8 seconds for a response.

  If no response, NNMi repeats this process using the next configured SNMP level.
- NNMi exhausts all possibilities. NNMi considers the hostname/IP-address to be a *non-SNMP* device until the next Discovery or Monitoring cycle.

Tip: It is best to use the same timeout/retry numbers for both ICMP and SNMP.

Your choice of SNMP Minimum Security Level determines the range of possibilities:

- If your SNMP Minimum Security Level is ***Community Only (SNMPv1)***, NNMi uses only SNMPv1 to locate SNMP agents.
- If your SNMP Minimum Security Level is ***Community Only (SNMPv1 or v2c)***, NNMi cycles through the following until successful:

  SNMPv2c

  SNMPv1

- If your SNMP Minimum Security Level is ***Community***, NNMi cycles through the following until successful:

  SNMPv2c

  SNMPv1

  SNMPv3 *No Authentication, No Privacy* settings (if any matching configurations, otherwise skip).

  SNMPv3 *Authentication, No Privacy* settings (if any matching configurations, otherwise skip).

  SNMPv3 *Authentication, Privacy* settings (if any matching configurations).

- If your SNMP Minimum Security Level is ***No Authentication, No Privacy***, NNMi cycles through the following until successful:

  SNMPv3 *No Authentication, No Privacy* settings (if any matching configurations at this, otherwise skip)

  SNMPv3 *Authentication, No Privacy* settings (if any matching configurations, otherwise skip).

  SNMPv3 *Authentication, Privacy* settings (if any matching configurations).

- If your SNMP Minimum Security Level is ***Authentication, No Privacy***, NNMi cycles through the following until successful:

  SNMPv3 *Authentication, No Privacy* settings (if any matching configurations, otherwise skip).

  SNMPv3 *Authentication, Privacy* settings (if any matching configurations).

- If your SNMP Minimum Security Level is **Authentication, Privacy**, NNMi cycles through the following until successful:

  SNMPv3 *Authentication, Privacy* settings (if any matching configurations).

Timeout / Retry Behavior Example for ICMP

When NNMi attempts to contact a device, your configuration settings for Timeout and Retry influence NNMi behavior.

NNMi attempts to contact the device using ICMP, then waits the configured timeout interval for a response. If not successful, NNMi increments the timeout interval before trying again. This process repeats until one of the following is true:

- The device responds to ICMP.
- The maximum configured number of ICMP Retries fails. NNMi considers the device unreachable through ICMP until the next Discovery or Monitoring cycle. For example, if your timeout is 2 seconds and your retry is 3:
  - NNMi attempts to communicate with a device and waits 2 seconds for a response.
  - If unsuccessful, NNMi retries and waits 4 seconds for a response.
  - If unsuccessful, NNMi retries a second time and waits 6 seconds for a response.
  - If unsuccessful, NNMi retries a third time and waits 8 seconds for a response.

Tip: It is best to use the same timeout/retry numbers for both ICMP and SNMP.

Configure Default Community Strings (SNMPv1 or SNMPv2c)

Use the Default Community Strings tab to provide default SNMPv1 and SNMPv2c community strings. For each address, NNMi checks the communication configuration settings in this order: communication protocols for Specific Nodes, communication protocols for Network Regions, and if no match is found, NNMi tries these default community strings. If NNMi discovers a device for which no SNMP settings are provided, that device is treated as a Non-SNMP device.

During initial discovery, NNMi tries many community strings until a match is found. After a match is identified for a node, the information is recorded to prevent future authentication errors.

Note: If you provide a read community string for a specific device, NNMi honors your choice and does not try any Region or Default community strings for that device.

NNMi uses SNMP read-only queries (Get commands) for ongoing discovery and monitoring of your network environment. SNMP read community strings are the validation passwords used to authenticate messages sent from NNMi to an SNMP agent. NNMi uses SNMP to gather useful information about the devices in your network environment. After receiving an SNMP request, an SNMP agent compares the read community string in the request to the read community strings that are configured for that SNMP agent. The SNMP agent responds to the request only when the request is accompanied by a valid community string.

During NNMi installation, any community strings that were provided are automatically stored in the table on the Default Community Strings tab.

Provide any number of additional community strings that are used broadly in your environment (for example, by default). The order in which your read community string settings appear in the table does not matter. NNMi checks all Default read community strings in parallel.

Tip: Having a large number of default community strings can negatively impact discovery performance. Instead of entering many default community strings, consider fine tuning the community string configuration for particular areas of your network by using the Regions or Specific Nodes settings.

NNMi uses the SNMPv2c settings to discover the SNMPv2c information about your network. This also determines whether NNMi *receives or discards incoming* SNMPv2c traps. Click here for more information.

- If the *incoming* trap's Source Node (and sometimes Source Object, such as card or interface) has not yet been discovered by NNMi, NNMi discards the trap. See "Handle Unresolved Incoming Traps" on page 776 for additional information. See also "Configure Network Devices to Send SNMP Notifications to NNMi" on page 771.
- If the Source Node was not discovered using SNMv3, NNMi discards any incoming SNMPv3 traps from that Node.
- NNMi discards traps that have no incident configuration or with an incident configuration set to Disabled. To ensure that NNMi retains all received Trap instances when your network environment includes SNMP agents using a variety of SNMPv1, SNMPv2c, and SNMPv3 protocol, you must configure two Incidents: one for the SNMPv1 version and one for the SNMPv2c/3 version of that trap. See "Configure SNMP Trap Incidents" on page 782.

- If either the Source Node or Source Object has *Management Mode* set to **Not Managed** or **Out of Service** in the NNMi database, NNMi always discards the incoming trap. See "Understand the Effects of Setting the Management Mode to Not Managed or Out of Service " on page 464.

  NNMi provides the Management Mode workspace so that you can quickly view lists of all nodes, interfaces, cards, addresses, or node components that NNMi is not currently discovering or monitoring. For information about these views:

- NNMi discards most incoming traps from network objects that are not monitored. For example, you can configure NNMi to exclude specified interfaces from being monitored. See "Monitoring Network Health" on page 340 for more information.

Note: If you want the NNMi management server to *forward* SNMPv2c traps to other machines in your network environment, see "Configure Trap Forwarding" on page 1376 for additional configuration steps.

To configure default SNMPv1 or SNMPv2c community strings for your environment:

1. Navigate to the **Communication Configuration** form.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Select the **Communication Configuration**.
2. Locate the **Default SNMPv1/v2c Community Strings** tab.
3. To provide a default *read community string*, navigate to the **Read Community Strings** table and do one of the following:
   - To establish a community string setting, click the New icon. In the Default Read Community String form, provide the required information (see table).
   - To edit a community string setting, click the Open icon in the row representing the community string setting you want to edit. In the Default Read Community String form, provide the required information (see table).
   - To delete a community string setting, select a row and click the Delete icon.
4. To provide a default *write community string*, navigate to **the Write Community String** attribute (see table).
5. Click **Save and Close** to return to the Communication Configuration form.
6. Click **Save and Close** to apply your changes.

Default SNMPv1 or SNMPv2c Community Strings

Attribute	Description
Read Community String	The SNMPv1 or SNMPv2c "Get" (read-only) Community String that is used as the default value for each SNMP Agent (case-sensitive). Many proxy vendors use the *read community string* for specifying remote target information. NNMi supports substitution parameters within read community strings for SNMPv1 or SNMPv2c proxy environments. Click here for more information. Copy and paste these codes at the end of your read community string to provide the

Default SNMPv1 or SNMPv2c Community Strings , continued

Attribute	Description
	values required by your proxy environment. NNMi substitutes the actual attribute values from the NNMi database at runtime: ${contextName} = Used for specifying VLAN context for switches (VLAN associated with the remote target node) ${managementAddress} = Node form, Management Address attribute value (the remote target node) ${snmpPort} = SNMP Agent form, UDP Port attribute value (SNMP agent associated with the remote target node) Type a maximum of 255 characters. Alpha-numeric, spaces, and special characters (~ ! @ # $ % ^ &amp; * ( ) _+ -) are permitted.
Ordering	*Optional*. During the Discovery process, NNMi tries Read Community Strings in priority order (lowest to highest). Then, NNMi tries all unordered Read Community Strings (treated as though they had the same Ordering number). These unordered requests are sent in parallel, with NNMi using the first response.
Write Community String	*Optional*. For use with the nnmsnmpset.ovpl command line tool The SNMPv1 or SNMPv2c "Set" (write) Community String that is used as the default value for each SNMP Agent (case-sensitive). **Tip**: SNMP Agents are often configured with different community strings for "Set" requests than for "Get" (read) requests. SNMPv1 and SNMPv2c require that you know the SNMP agent's *write community string* before you can change settings on any device. The nnmsnmpset.ovpl command can use the value you provide here, rather than requiring that you type the write community string each time you invoke the command. Type a maximum of 255 characters. Alpha-numeric, spaces, and special characters (~ ! @ # $ % ^ &amp; * ( ) _+ -) are permitted. Because this is a type of password, you must enter the value twice.

Default Read Community String Form

For each IP address, NNMi checks the communication configuration settings in this order: communication protocols for Specific Devices, communication protocols for Network Regions, and if no match is found, NNMi tries the default community strings. If NNMi discovers a device for which no community string is provided, that device is treated as a Non-SNMP device.

To provide a default community string for your environment:

1. Navigate to the **Default Read Community String** form.

   a. From the workspace navigation panel, select the **Configuration** workspace.

   b. Select the **Communication Configuration**.

   c. Navigate to the **Default SNMPv1/v2c Community Strings** tab.

   d. Navigate to the **Read Community Strings** table.

   e. Do one of the following:

      - To establish a community string setting, click the New icon.

      - To edit a community string setting, select a row, click the Open icon in the row representing the configuration you want to edit.

2. Provide the read community string (see table).

   Provide any number of additional SNMPv1 or SNMPv2c read community strings that are used broadly in your environment (for example, by default).

3. Click either:

   - **Save and Close** to return to the Communication Configuration form.

   - Save and New to add another community string.

4. Click **Save and Close** to apply your changes.

To determine which Community Strings are relevant for a node, select the node in an NNMi map or table view, and click Actions → Configuration Details → Communication Settings. In the Communities list, Ordering number is in parentheses. For example: communityString (200).

Default Read Community String

Attribute	Description
Read Community String	The SNMP "Get" (read-only) Community String that is used in your network environment (case-sensitive). Many proxy vendors use the *read community string* for specifying remote target information. NNMi supports substitution parameters within read community strings for SNMPv1 or SNMPv2c proxy environments. Click here for more information. Copy and paste these codes at the end of your read community string to provide the values required by your proxy environment. NNMi substitutes the actual attribute values from the NNMi database at runtime: ${contextName} = Used for specifying VLAN context for switches (VLAN associated with the remote target node) ${managementAddress} = Node form, Management Address attribute value (the remote target node) ${snmpPort} = SNMP Agent form, UDP Port attribute value (SNMP agent associated with the remote target node) Type a maximum of 255 characters. Alpha-numeric, spaces, and special characters (~ ! @ # $ % ^ &amp; * ( ) _+ -) are permitted.
Ordering	*Optional.* A numeric value. NNMi uses the first Community String that results in successful SNMP communication:

Default Read Community String , continued

Attribute	Description
	• Each ordering number must be unique (no duplicate numbers). NNMi tries the provided Community Strings in the order you define (lowest number first). **Tip**: Consider incrementing by 10s or 100s to provide flexibility when adding new Read Community Strings over time. • If no Ordering numbers are specified, NNMi tries all community strings in parallel. • If some but not all the community strings have an Ordering number, NNMi tries the community strings with a specified Ordering number first. Then, NNMi tries all the community strings without an Ordering number in parallel.

Configure Default SNMPv3 Settings

Use the Default SNMPv3 Settings tab to provide default SNMPv3 user-based security model (USM) settings. For each address, NNMi checks the communication configuration settings in this order: communication protocols for Specific Nodes, communication protocols for Network Regions, and if no match is found, NNMi tries these default user-based security model (USM) settings. If NNMi discovers a device for which no SNMP settings are provided, that device is treated as a Non-SNMP device.

During initial discovery, NNMi tries many SNMP configuration settings until a match is found. After a match is identified for a Node, the information is recorded to prevent future authentication errors.

Note: If you provide SNMPv3 user-based security model (USM) settings for a specific device, NNMi honors your choice and does not try any Region or Default settings for that device.

NNMi uses SNMP queries for ongoing discovery and monitoring of your network environment. SNMPv3 user-based security model (USM) settings are used to authenticate messages sent from NNMi to an SNMP agent. NNMi uses SNMP to gather useful information about the devices in your network environment. After receiving an SNMP request, an SNMP agent compares the SNMPv3 user-based security model (USM) settings in the request to the SNMPv3 user-based security model (USM) settings that are configured for that SNMP agent. The SNMP agent responds to the request only when the request is accompanied by valid SNMPv3 user-based security model (USM) settings.

Provide any number of additional SNMPv3 user-based security model (USM) settings that are used broadly in your environment (for example, by default). The order in which your SNMPv3 user-based security model (USM) settings appear in this table does not matter. NNMi checks all Default SNMPv3 Settings at a particular security level in parallel.

NNMi uses Default SNMPv3 user-based security model (USM) settings to access devices.

To view the current list of default SNMPv3 USM settings:

1. Navigate to the **Default SNMPv3 Settings** tab.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Select **Communication Configuration**.

c. Navigate to the **Default SNMPv3 Settings** tab.

2. The displayed table lists the Unique Name of each default SNMPv3 USM setting.

   Note: NNMi tries to use the Specific Node SNMPv3 Settings. If none match, NNMi tries the Region SNMPv3 Settings. If none match, NNMi tries the default SMNPv3 settings provided here.

3. You can do the following:

   - To establish a new setting, click the ✳ New icon. See "Default SNMPv3 Settings form" below.

     Click **Save and Close** to return to the Default SNMPv3 Settings form.

   - To edit an existing setting, select a row, click the Open icon. See "Default SNMPv3 Settings form" below.

     Click **Save and Close** to return to the Default SNMPv3 Settings form.

   - To delete an existing setting from the Default list, select a row and click the ✖ Delete icon.

     Note: The record remains in the database for possible use elsewhere and is simply removed from the Default list.

4. Click **Save and Close** to return to the Communication Configuration form.

5. Click **Save and Close** to apply your changes.

Default SNMPv3 Settings form

NNMi can use SNMPv3 user-based security model (USM) settings to access devices.

NNMi tries to use the current SNMPv3 Settings attribute value from Specific Node Settings. If none match, NNMi tries the Region SNMPv3 Settings. If none match, NNMi tries the default SMNPv3 settings provided here.

To configure a Default SNMPv3 Setting:

1. Navigate to the **Default SNMPv3 Settings** form.

   a. From the workspace navigation panel, select the **Configuration** workspace.

   b. Select the **Communication Configuration**.

   c. Navigate to the **Default SNMPv3 Settings** tab.

   d. Do one of the following:

      ○ To create default SNMPv3 Setting definition, click the ✳ New icon.

      ○ To edit a default SNMPv3 Setting, select a row, click the Open icon.

2. Click the SNMPv3 Settings ▾ Lookup icon and select one of the options from the drop-down menu:

   - Show Analysis to display Analysis Pane information for the currently configured (selected) SNMPv3 Setting name. (See Use the Analysis Pane for more information about the Analysis Pane.)

- Quick Find to view and select from the list of all existing SNMPv3 Settings (for more information see "Use the Quick Find Window" on page 41).
- Open to display the details of the currently configured (selected) SNMPv3 Setting (see "SNMPv3 Settings Form" for more information).
- New to create a new SNMPv3 Setting (see "SNMPv3 Settings Form" for more information).

3. Click **Save and Close** to return to the Default SNMPv3 Settings form.
4. Click **Save and Close** to return to the Communication Configuration form.
5. Click **Save and Close** to apply your changes.

Configure the Default Device Credentials

NNMi uses the Device Credentials settings for the following:

- Device discovery of some vendor-specific devices that require non-SNMP communication, such as Netconf over SSH. For a list of the these devices see the NNMi Device Support Matrix.
- *HP Network Node Manager iSPI Network Engineering Toolset Software*

NNMi uses the following sequence to determine Device Credentials:

- Use the Specific Node Device Credentials. If none match, continue.
- Use the Region Device Credentials. If none match, continue.
- Use the Default Credential settings (provided here).

To provide the default credentials setting:

1. Navigate to the **Default Device Credentials** tab.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Select **Communication Configuration**.
   c. Navigate to the **Default Device Credentials** tab.
2. Provide the default attribute values (see table).
3. Click **Save and Close** to return to the Communication Configuration form.
4. Click **Save and Close** to apply your changes.

Note: *HP Network Node Manager iSPI Network Engineering Toolset Software* uses the Default Credentials setting to access devices when running Diagnostics either automatically or when the **Actions → Run Diagnostics (iSPI NET only)** option is used. (See "Configure Diagnostics for an Incident (NNM iSPI NET)" on page 757 and Node Form: Diagnostics Tab for more information.)

> **Tip:** You can right-click any object in a table or map view to access the **Actions** menu.

At each level in the sequence to determine the Device Credentials (see bullet list above), NNMi first uses Secure Shell (SSH) to establish a secure connection, and if the SSH attempt fails, NNMi tries Telnet protocol as the communication method.

> **Caution:** By default, neither Microsoft Internet Explorer nor Mozilla Firefox defines the telnet command nor the SSH command, so using either of these menu items produces an error message. See the "Configuring the Telnet and SSH Protocols for Use by NNMi" chapter in the *HP Network Node Manager i Software Deployment Reference* for configuration information.

Default Device Credential Attributes

Attribute	Description
User Name	Type the user name that you want NNMi to use for logging into devices by default (when no Region or Specific Node settings work).
Password	Type the password that you want NNMi to use for logging into devices by default (when no Region or Specific Node settings work). **Note:** NNMi encrypts the password and displays asterisks for this attribute. If you want to change the password, first clear the asterisks displayed in the Password attribute and enter the new Password value.

Configure Regions (Communication Settings)

Configuring communication protocols for regions is optional.

Note: If you provide an SNMPv1 or SNMPv2c *read community string* or an SNMPv3 USM Setting for a specific device, NNMi honors your choice and does not try any Region or Default settings for that device.

Use the Regions tab to fine tune communication protocol usage and settings for particular regions of your network (for example, buildings, floors within those buildings, workgroups within a particular floor, or **private IP addresses**[1]). When you leave a field blank in a region definition, NNMi uses the next applicable configuration setting in the following order:

- The value for each field as defined in the first Region definition that matches, Regions are checked according to the Ordering number. The match with the lowest Ordering number applies.
- If no Region definition provides a value for an attribute, the default value is used.

[1] These are IPv4 addresses that can be reused in home and office local area networks (LANs). Following the standards set by RFC 1918 and RFC 4193 (10.*.*.*, 169.254.*.*, 172.16-31.*.*, and 192.168.*.*)

Note: NNMi enables you to set up one or more SNMP Proxy Servers when an SNMP node is otherwise unreachable (for example, when a node you want to manage is behind a firewall). To enable NNMi to use the SNMP Proxy Server, when you configure communication protocols for network regions, you must include the IP address and port number on the SNMP Proxy Server. See "Communication Region Form" below for more information.

If your communication protocol usage is too complex for Region definitions, see "Configure Specific Nodes (Communication Settings)" on page 154.

To configure communication protocols for a particular region of your network:

1. Navigate to the **Communication Region** form.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Select the **Communication Configuration**.
   c. Navigate to the **Regions** tab.
   d. Do one of the following:
      - To establish a region definition, click the New icon, and continue.
      - To edit a region definition, select a row, click the Open icon, and continue.
      - To delete a region definition, select a row and click the Delete icon.
2. Provide the required information. Define the regions with wildcard address, wildcard device names, or literal addresses and names . See "Communication Region Form" below.
3. Click **Save and Close** to return to the Communication Configuration form.
4. Click **Save and Close** to apply your changes.

Related Topics:

"Configure Default SNMP, Management Address, and ICMP Settings" on page 120

"Configure Default Community Strings (SNMPv1 or SNMPv2c)" on page 129

"Configure Specific Nodes (Communication Settings)" on page 154

Communication Region Form

To configure communication regions:

1. Navigate to the **Communication Region** form.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Select the **Communication Configuration**.
   c. Navigate to the **Regions** tab.
   d. Do one of the following:
      - To establish a region definition, click the New icon.
      - To edit a region definition, select a row, click the Open icon.

2. Provide the basic communication region definition (see the Regional Basic Settings table, Regional SNMP Settings table, and Regional ICMP Settings table).
3. Make your configuration choices. Click here for a list of choices .
4. Click **Save and Close** to return to the Communication Configuration form.
5. Click **Save and Close** to apply your changes.

Regional Basic Settings

Attribute	Description
Name	A name for this region.
Ordering	A numeric value. NNMi checks for configuration settings in the order you define (lowest number first). NNMi uses the first match found for each address. No duplicate Ordering numbers are permitted. Each Communication Region ordering number must be unique. **Tip:** Consider incrementing Ordering numbers by 10s or 100s to provide flexibility when adding new regions over time.
Description	*Optional*. Provide any description that would be useful for communication purposes within your team. Type a maximum of 255 characters. Alpha-numeric, spaces, and special characters (~ ! @ # $ % ^ & * ( ) _+ -) are permitted.

Regional SNMP Settings

Attribute	Description
Enable SNMP Communication	If ☑ enabled, the Discovery Process and State Poller Service generate network traffic with SNMP protocol to discover and monitor your network devices in this region. If □ disabled, NNMi does not generate any SNMP traffic on your network in this region. **Caution:** At least one IP Address in each node must have SNMP enabled, otherwise no SNMP data is collected from that Node. With no SNMP data, Spiral Discovery interprets each IP Address as a separate node, Causal Engine calculates Status based only on IP address State, previously discovered Interfaces show a State attribute value of "Not Polled" and a Status attribute value of "No Status" with the Interface map-symbol color set to beige, and no new Interfaces are discovered. **Note:** If you use Auto-Discovery, NNMi might detect Nodes and add them to the NNMi database as non-SNMP nodes. To configure Auto-

Regional SNMP Settings, continued

Attribute	Description
	Discovery to not add specified IP addresses to the NNMi database, not acknowledge any Hints received about them, nor gather Discovery Hints from them unless the address is a discovery seed, see "Set Outside Limits for Auto-Discovery" on page 228.
Enable SNMP Address Rediscovery	**Note:** The NNMi administrator can over-ride this setting on a per-node basis. See "Specific Node Settings Form (Communication Settings)" on page 155. If ☑ enabled, NNMi automatically identifies which management address (SNMP agent) to use for each device. If the initially configured address becomes unreachable, NNMi automatically locates another address, if possible, and changes the management address attribute value. Click here for more information. When NNMi first discovers a node, the *seed address* (provided by the NNMi administrator) or discovered address (for non-seeded nodes) becomes the initial address used for SNMP communication. After NNMi builds an inventory of all IP addresses associated with the node (see "How Spiral Discovery Works" on page 176), NNMi follows a set of rules to determine which address is the best choice for each node's Management Address: **Note:** With NNMi Advanced, the NNMi administrator specifies whether NNMi prefers IPv4 or IPv6 addresses when selecting the Management Address. See Configure Default SNMP, Management Address, and ICMP Settings. 1. NNMi ignores the following addresses when determining which Management Address is most appropriate: ▪ Any address of an administratively-down interface. ▪ Any address that is virtual (for example, **VRRP**[1]). ▪ Any IPv4 **Anycast Rendezvous Point IP Address**[2] or IPv6 Anycast address. ▪ Any address in the reserved loopback network range. IPv4 uses 127/24 (`127.*.*.*`) and IPv6 uses `::1`.

[1] Virtual Router Redundancy Protocol

[2] Rendezvous Point addresses are loopback addresses used for routers in multi-cast network configurations.

Regional SNMP Settings, continued

Attribute	Description
	▪ Any **IPv6 link-local address**[1]. 2. If the NNMi Administrator specifies Enable SNMP Address Rediscovery in Communication Configuration, NNMi prefers the last-known Management Address (if any). 3. If the Management Address does not respond and the NNMi Administrator specifies Enable SNMP Address Rediscovery in Communication Configuration, NNMi uses the Communication Configuration settings for *Management Address Selection*. The NNMi Administrators chooses the order in which NNMi checks the following: ▪ Seed IP / Management IP - If the NNMi Administrator configures a Seed, NNMi uses the Seed address (either a specified IP address or the DNS address associated with a specified hostname) only during initial Discovery. NNMi then requests the current Management Address (the address from which the node's SNMP Agent responds) and uses that IP address for all communication after initial discovery. ▪ Lowest Loopback - If a node supports multiple **loopback address**[2], NNMi queries each loopback addresses, starting with the lowest number. NNMi uses the loopback address with the lowest number from which the SNMP agent responds (for example, 10.16.42.197 is a lower number than 10.16.197.42). ▪ Highest Loopback - If a node supports multiple **loopback address**[3], NNMi queries each loopback addresses, starting with the highest number. NNMi uses the loopback address with the highest number from which the SNMP agent responds. ▪ Interface Matching - The NNMi Administrator chooses which interface MIB variable NNMi queries to detect changes. NNMi can use the following MIB-II attribute values: `ifIndex`, `ifName`, `ifDescr`, `ifAlias`, or a combination of these (`ifName` or `ifDescr`, `ifName` or `ifDescr` or `ifAlias`).

[1] A non-routable IPv6 unicast address only used for communication with other nodes on the same link (LAN or VLAN). Link local addresses cannot be used for communication that must be forwarded through a router. IPv6 auto-configuration automatically assigns a unique link local address in the fe80::/10 address space to each IPv6-enabled interface on a system.

[2] The address associated with the loopback interface. The loopback interface is a virtual interface on a device that provides a route for internal communication. Many vendors provide a specially configured loopback for management purposes. Exact details of how loopbacks are configured varies by vendor and model. See each device's documentation for details. NNMi identifies these loopback addresses by using ifType Number 24, softwareloopback from the IANA ifType-MIB.

[3] The address associated with the loopback interface. The loopback interface is a virtual interface on a device that provides a route for internal communication. Many vendors provide a specially configured loopback for management purposes. Exact details of how loopbacks are configured varies by vendor and model. See each device's documentation for details. NNMi identifies these loopback addresses by using ifType Number 24, softwareloopback from the IANA ifType-MIB.

Regional SNMP Settings, continued

Attribute	Description
	4. If no response, NNMi queries any remaining IP addresses in the node's IP address inventory, starting with the lowest number. NNMi uses the address with the lowest number from which the SNMP agent responds. 5. If no response, NNMi checks for any Mapped Address configured for one of the currently known addresses (see the Mapped Address column in the Custom IP Addresses view). **Note:** The address represents a *static* Network Address Translation (NAT) pair's *external IP address* from the internal/external IP address pair. NNMi Administrators configure these pairs using the Overlapping IP Address Mapping form. NNMi uses this list of addresses starting with IPv4 from low to high, then IPv6 from low to high. 6. If no response, NNMi might be configured to repeat the sequence using SNMPv1, SNMPv2c, or SNMPv3 in the order specified by the NNMi administrator (Communication Configurations *SNMP Minimum Security Level* settings). 7. When all else fails, NNMi retains the last known Management Address (if any) and automatically changes the State of that SNMP Agent object to Critical. This process is repeated during each Spiral Discovery cycle, and the Management Address can change. For example, NNMi's inventory of addresses for the node expands, or the current Management Address does not respond to SNMP queries due to network problems or node reconfiguration. The NNMi administrator can prevent changes to the management address using the Communication Configurations *Enable SNMP Address Rediscovery* or *Preferred Management Address* setting. If ☐ disabled, when the current management address (SNMP agent) becomes unreachable, NNMi reclassifies the node as a non-SNMP node until the previously configured management address is available again.
Enable SNMP GetBulk	*Applies only to SNMPv2 or higher.* If you have devices in your network environment that have trouble responding to `GetBulk` commands, you can instruct NNMi to use `Get` or `GetNext` instead of `GetBulk`. If ☑ enabled, NNMi uses the SNMPv2c `GetBulk` command to gather information from devices in this Region of your network environment. If ☐ disabled, NNMi uses the SNMP `Get` or `GetNext` command to gather information from devices in this Region of your network environment (requesting responses for one SNMP OID at a time).
SNMP Timeout	(Seconds:Milliseconds) Maximum 1 millisecond less than a minute: 59 seconds 999 milliseconds.

Regional SNMP Settings, continued

Attribute	Description
	Time that NNMi waits for a response to an SNMP query before reissuing the request. Both the Discovery Process and the State Poller Service use this setting in this region. For an explanation of how NNMi implements timeout and retry configurations, see "Timeout / Retry Behavior Example for SNMP" on page 127.
SNMP Retries Count	Maximum number of retries that NNMi issues for an SNMP query before determining the query result to be "unresponsive". Zero means no retries. Both the Discovery Process and the State Poller Service use this setting in this region.
SNMP Port	Default is 161. Specifies the management server's port that NNMi uses when generating SNMP traffic. Both the Discovery Process and the State Poller Service use this setting in this region.
SNMP Proxy Address	*Optional*. IP address of the your SNMP Proxy Server (for example, a proxy that gathers data from non-SNMP devices and can use that data to respond to NNMi SNMP requests). To enable a proxy, you must also provide the port number of your SNMP Proxy Server. See SNMP Proxy Port (next attribute). **Note:** When you configure NNMi to use a Proxy Server, you must ensure that the Proxy Server vendor supports the Object Identifiers used to handle SNMP requests and responses. See the “NNMi Communications” chapter of the *HP Network Node Manager i Software Deployment Reference* for more information.
SNMP Proxy Port	*Optional*. Port number of the SNMP Proxy Server. To enable a proxy, you must also provide the IP address of your SNMP Proxy Server. See SNMP Proxy Address (previous attribute). **Note:** When you configure NNMi to use a Proxy Server, you must ensure that the Proxy Server vendor supports the Object Identifiers used to handle SNMP requests and responses. See the “NNMi Communications” chapter of the *HP Network Node Manager i Software Deployment Reference* for more information.
SNMP Minimum Security Level	This setting determines whether each NNMi Rediscovery cycle automatically detects the best SNMP choice (v1, v2, or v3) for each Node (automatically detects any upgrade to the SNMP agent on each Node), or uses only the SNMP version that you specify. For SNMPv1 or SNMPv2c, configure NNMi to use Community Strings in your network environment: • Community Only (SNMPv1)

Regional SNMP Settings, continued

Attribute	Description
	NNMi tries only SNMPv1 settings. • Community Only (SNMPv1 or v2c) NNMi first tries to use SNMPv2c settings, and, if that fails, NNMi tries SNMPv1 settings. • Community NNMi first tries to use SNMPv2c settings, and, if that fails, NNMi tries SNMPv1 settings. If both SNMPv2c and SNMPv1 fail, NNMi tries SNMPv3 settings if any are available. For SNMPv3, configure NNMi to use the User-based Security Module (USM) level of security required in your network environment (if your environment also uses SNMPv1/SNMPv2c, select Community): • No Authentication, No Privacy • Authentication, No Privacy • Authentication, Privacy See"Timeout / Retry Behavior Example for SNMP" on page 127 for an explanation of NNMi behavior with each of these choices.

Regional ICMP Settings

Attribute	Description
Enable ICMP Communication	If ☑ enabled, NNMi generates network traffic with ICMP protocol in this region. If □ disabled, NNMi does not generate any ICMP traffic on your network in this region: • Addresses in this Region (both previously discovered and newly discovered) have a State attribute value of "Not Polled" and a Status attribute value of "No Status" with the IP Address map-symbol color set to beige. • Nodes with all IP addresses and interfaces showing a Status attribute value of "No Status" have a map-symbol background shape color set to beige. However, it is possible for a node to have IP addresses in multiple regions with multiple Status values. **Note:** See "Monitoring Network Health" on page 340 for information about enabling/disabling ICMP communication specifically for the State Poller Service.
ICMP Timeout	(Seconds:Milliseconds) Maximum 1 millisecond less than a minute: 59 seconds 999 milliseconds. Time that NNMi waits for a response to an ICMP query before reissuing the

Regional ICMP Settings, continued

Attribute	Description
	request in this region. For an explanation of how NNMi implements timeout and retry configurations, see "Timeout / Retry Behavior Example for ICMP" on page 128.
ICMP Retries Count	Maximum number of retries that NNMi issues for an ICMP query in this region before logging an error. Zero means no retries.

Configure Address Ranges for Regions

To configure an address range for this region:

1. Navigate to the **Region Included Address Range** form.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Select **Communication Configuration**.
   c. Navigate to the **Regions** tab.
   d. Do one of the following:
      - To establish a region definition, click the ✱ New icon.
      - To edit a region definition, select a row, click the Open icon.
   e. In the **Communication Region** form, navigate to the **Included Address Regions** tab.
   f. Do one of the following:
      - To establish an address range setting, click the ✱ New icon.
      - To edit an address range setting, select a row, click the Open icon.
      - To delete an address range setting, select a row and click the ✖ Delete icon.
2. Provide address range definition (see table).

   If you provide multiple IP address ranges for a region, each device must pass at least one to meet the criteria.

   Tip: If you provide both IP address ranges and hostname wildcards, each device must pass at least one in either category (not both) to meet the criteria.
3. Click **Save and Close** to return to the Communication Region form.
4. Click **Save and Close** to return to the Communication Configuration form.
5. Click **Save and Close** to apply your changes.

Address Range Definition Attribute

Attribute	Description
IP Range	To specify a range of IP addresses for this Communications Region, use one of the

Address Range Definition Attribute , continued

<table>
<tr><th>Attribute</th><th>Description</th></tr>
<tr><td></td><td>
<p>following. Pick one address notation style, combinations of wildcards and CIDR notation are not permitted within one address range. You can provide multiple address range settings:</p>
<ul>
<li><p><b>IPv4 address wildcard notation</b>.</p>
<p>An IPv4 Address range is a modified dotted-notation where each octet is one of the following:</p>
<ul>
<li>A specific octet value between 0 and 255</li>
<li>A low-high range specification for the octet value (for example, "112-119")</li>
<li>An asterisk (*) wildcard character, which is equivalent to the range expression "0-255"</li>
</ul>
<blockquote><b>Note:</b> The following two IPv4 addresses are considered invalid: <code>0.0.0.0</code> and <code>127.0.0.0</code></blockquote>
<p>Examples of valid IPv4 address wildcards include:</p>
<pre>10.1.1.*
10.*.*.*
10.1.1.1-99
10.10.50-55.*
10.22.*.4 10.1-9.1-9.1-9</pre>
</li>
<li><p><b>IPv4 Classless Inter-Domain Routing (CIDR) notation</b>.</p>
<p>The CIDR notation specifies the number of consecutive bits in the IPv4 address that must match.</p>
<p>For example, <code>10.2.120.0/21</code></p>
<blockquote><b>Note:</b> NNMi does not support CIDR subnet mask notation such as, <code>10.2.120.0/255.255.248.0</code></blockquote>
<table>
<tr><th>Example IPv4 Prefix Length Values</th><th>Number of Usable IPv4 Addresses</th></tr>
<tr><td>28</td><td>14 (16-2=14)*</td></tr>
<tr><td>29</td><td>6 (8-2=6)*</td></tr>
<tr><td>30</td><td>2 (4-2=2)*</td></tr>
<tr><td>31</td><td>2</td></tr>
</table>
<p>* Two IPv4 addresses are reserved in each subnet. The first IPv4 address is used for the network itself and the last IPv4 address is reserved for broadcast.</p>
</li>
<li><p><b>IPv6 address wildcard notation</b></p></li>
</ul>
</td></tr>
</table>

Address Range Definition Attribute , continued

Attribute	Description

	Separate each 16-bit value of the IPv6 address with a colon. The 16-bit value can be any of the following:
	■ A specific hexadecimal value between `0` and `FFFF` (case insensitive).
	■ A low-high range specification of the hexadecimal value (for example, `1-1fe`).
	■ An asterisk (`*`) wildcard character (equivalent to the range expression `0-ffff`).
	Note: The standard IPv6 short-hand notation (`::`) is allowed to express one or more 16-bit elements of zero (`0`) values. However, the mixed IPv6/IPv4 dot-notation (for example, `2001:d88::1.2.3.4`) is not permitted as an IPv6 address range.
	Valid examples of ranges in modified IPv6 address notation include the following:
	`2001:D88:0:A00-AFF:*:*:*:*`
	`2001:D88:1:*:*:*:*:*`
	`2001:D88:2:0:a07:ffff:0a01:3200-37ff`
	• **IPv6 Classless Inter-Domain Routing (CIDR) notation**
	The CIDR notation specifies the number of consecutive bits in the IPv6 address that must match.
	`2001:d88:a00::/44` (equivalent to modified IPv6 address notation `2001:d88:a00-a0f:*:*:*:*:*`)
	For example, valid IPv6 address ranges in CIDR notation include the following:
	`2001:d88:0:a00::/56` (equivalent to modified IPv6 address notation `2001:D88:0:A00-AFF:*:*:*:*`)
	`2001:d88:1::/48` (equivalent to modified IPv6 address notation `2001:D88:1:*:*:*:*:*`)

Configure Hostname Filters for Regions

Define the Communication Region with hostname patterns.

To establish a Hostname Filter setting:

1. Navigate to the **Region Hostname Filter** form.
   - From the workspace navigation panel, select the **Configuration** workspace.
   - Select the **Communication Configuration**.
   - Navigate to the **Regions** tab.
   - Do one of the following:

        - To create a region definition, click the New icon.
        - To edit a region definition, select a row, click the Open icon.
    - In the **Communication Region** form, access the **Hostname Filters** tab.
    - Do one of the following:
        - To create a hostname wildcard definition, click the New icon.
        - To edit a hostname wildcard definition, select a row, click the Open icon.
        - To delete a hostname wildcard setting, select a row and click the Delete icon.

2. Type an appropriate hostname filter (see table).

   If you provide multiple hostname wildcard expressions for a region, each device must pass at least one to meet the criteria for the Region.

   Tip: If you provide both hostname wildcards and IP address ranges, each device must pass at least one in either category (not both) to meet the criteria for the Region.

3. Click **Save and Close** to return to the Communication Region form.
4. Click **Save and Close** to return to the Communication Configuration form.
5. Click **Save and Close** to apply your changes. See "Discovering Your Network" on page 175 and Verify Device Configuration Details.

Node Hostname Filter Definition

Attribute	Description
Hostname Filter	Enter a wildcard expression using the following characters as wildcards: • ? = one character • * = multiple characters Wildcard expressions are *not case-sensitive*. So a wildcard of ABC* would match devices with hostnames beginning with ABC*, abc*, and Abc* **Caution**: The Hostname attribute value on the Node form of the discovered node must match (not case-sensitive) what is entered here. NNMi follows a set of rules to dynamically generate the value stored in the NNMi database for each Node's Hostname. Click here for details. • If the Node supports SNMP, NNMi requests the Hostname using the IP Address of the associated SNMP agent (the Management Address attribute value on the Node form). When the NNMi administrator chooses **Enable SNMP Address Rediscovery** ☑ in the Communication Configuration: ▪ If the SNMP Agent does not respond, NNMi checks for another Management Address to request the Hostname, and the Hostname could change.

Node Hostname Filter Definition , continued

Attribute	Description
	▪ If the SNMP Agent associated with the node changes, the Management Address and Hostname could change. When the NNMi administrator disables **Enable SNMP Address Rediscovery** ☐ in the Communication Configuration: ▪ If the SNMP Agent does not respond, NNMi uses the previously gathered Management Address attribute value to request the Hostname. ▪ If the SNMP Agent associated with the node changes, NNMi uses the previously gathered Management Address attribute value to request the Hostname. • If the Node does not support SNMP, no Management Address is available. NNMi requests a Hostname starting with the lowest IP Address associated with the node (a Discovery Seed value or an IP address value gathered from a neighboring device). NNMi uses the first Hostname provided. The Hostname might change during a future discovery cycle. **Note:** NNMi administrators can use NNMi property file settings to change the way NNMi determines Hostname values: • `nms-topology.properties` file settings: If DNS is the source of the Node's Hostname, there are three choices. By default NNMi uses the exact Hostname from your network configuration. It is possible to change NNMi behavior to convert Hostnames to all uppercase or all lowercase. See the "Modifying NNMi Normalization Properties" section of the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`. • `nms-disco.properties` file settings: The Hostname is either requested from the Node's lowest loopback interface IP address that resolves to a Hostname or requested from the Node's designated Management Address (SNMP agent address). With either choice, when no IP address resolves to a Hostname, the IP address itself becomes the Hostname. See the "Maintaining NNMi" chapter of the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.

Configure SNMPv1/v2c Community Strings for Regions

If more than one SNMPv1 or SNMPv2c "get" community string is used within this region, repeat this step any number of times. Order does not matter because all community strings defined for this Region are checked in parallel.

During initial discovery, NNMi tries many community strings until a match is found. After a match is identified for a Node, the information is recorded to prevent future authentication errors.

NNMi uses the SNMPv2c settings to discover the SNMPv2c information about your network. This also determines whether NNMi *receives or discards incoming* SNMPv2c traps. Click here for more information.

- If the *incoming* trap's Source Node (and sometimes Source Object, such as card or interface) has not yet been discovered by NNMi, NNMi discards the trap. See "Handle Unresolved Incoming Traps" on page 776 for additional information. See also "Configure Network Devices to Send SNMP Notifications to NNMi" on page 771.

- If the Source Node was not discovered using SNMv3, NNMi discards any incoming SNMPv3 traps from that Node.

- NNMi discards traps that have no incident configuration or with an incident configuration set to Disabled. To ensure that NNMi retains all received Trap instances when your network environment includes SNMP agents using a variety of SNMPv1, SNMPv2c, and SNMPv3 protocol, you must configure two Incidents: one for the SNMPv1 version and one for the SNMPv2c/3 version of that trap. See "Configure SNMP Trap Incidents" on page 782.

- If either the Source Node or Source Object has *Management Mode* set to **Not Managed** or **Out of Service** in the NNMi database, NNMi always discards the incoming trap. See "Understand the Effects of Setting the Management Mode to Not Managed or Out of Service " on page 464.

  NNMi provides the Management Mode workspace so that you can quickly view lists of all nodes, interfaces, cards, addresses, or node components that NNMi is not currently discovering or monitoring. For information about these views:

- NNMi discards most incoming traps from network objects that are not monitored. For example, you can configure NNMi to exclude specified interfaces from being monitored. See "Monitoring Network Health" on page 340 for more information.

Note: If you want the NNMi management server to *forward* SNMPv2c traps to other machines in your network environment, see "Configure Trap Forwarding" on page 1376 for additional configuration steps.

To provide a community string for this region:

1. Navigate to the **Communication Region** form.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Select **Communication Configuration**.
   c. Navigate to the **Regions** tab.
   d. Do one of the following:
      - To establish a region definition, click the ✱ New icon.
      - To edit a region definition, select a row, click the Open icon.
2. In the **Communication Region** form, navigate to the **SNMPv1/v2c Community Strings** tab.
3. To provide a *read community string*, navigate to the **Read Community Strings** table and do one of the following:

   Note: If you do not provide any community strings, NNMi uses the Default Community Strings.

- To establish a community string setting, click the New icon, and provide the required information (see table).
- To edit a community string setting, select a row, click the Open icon, and provide the required information (see table)
- To delete a community string setting, select a row and click the Delete icon

4. To provide a *write community string* for this region, navigate to **the Write Community String** attribute (see table).

   Note: If you do not provide any community strings, NNMi uses the Default Community Strings.

5. Click **Save and Close** to return to the Communication Region form.
6. Click **Save and Close** to return to the Communication Configuration form.
7. Click **Save and Close** to apply your changes.

SNMPv1 or SNMPv2c Community String for this Region

Attribute	Description
Read Community String	The SNMPv1 or SNMPv2c "Get" (read-only) Community String that is used for this region (case-sensitive). **Tip**: If no values appear in this table, the default settings are used (see "Configure Default Community Strings (SNMPv1 or SNMPv2c)" on page 129). Many proxy vendors use the *read community string* for specifying remote target information. NNMi supports substitution parameters within read community strings for SNMPv1 or SNMPv2c proxy environments. Click here for more information. Copy and paste these codes at the end of your read community string to provide the values required by your proxy environment. NNMi substitutes the actual attribute values from the NNMi database at runtime: ${contextName} = Used for specifying VLAN context for switches (VLAN associated with the remote target node) ${managementAddress} = Node form, Management Address attribute value (the remote target node) ${snmpPort} = SNMP Agent form, UDP Port attribute value (SNMP agent associated with the remote target node) Type a maximum of 255 characters. Alpha-numeric, spaces, and special characters (~ ! @ # $ % ^ &amp; * ( ) _+ -) are permitted.
Ordering	*Optional*. During the Discovery process, NNMi tries Read Community Strings in priority order (lowest to highest). Then, NNMi tries all unordered Read Community Strings (treated as though they had the same Ordering number). These unordered requests are sent in parallel, with NNMi using the first response.
Write Community	*Optional*. For use with the nnmsnmpset.ovpl command line tool. The SNMPv1 or SNMPv2c "Set" (write) Community String that is used for the

SNMPv1 or SNMPv2c Community String for this Region , continued

Attribute	Description
String	SNMP Agent for each node in this region (case-sensitive). **Tip**: SNMP Agents are often configured with different community strings for "Set" requests than for "Get" (read) requests. SNMPv1 and SNMPv2c require that you know the SNMP agent's *write community string* before you can change settings on any device. The nnmsnmpset.ovpl command can use the value you provide here, rather than requiring that you type the write community string each time you invoke the command. **Tip**: If no value is provided here, the default settings are used (see "Configure Default Community Strings (SNMPv1 or SNMPv2c)" on page 129). Type a maximum of 255 characters. Alpha-numeric, spaces, and special characters (~ ! @ # $ % ^ &amp; * ( ) _+ -) are permitted. Because this is a type of password, you must enter the value twice.

Configure SNMPv3 Settings for Regions

NNMi can use SNMPv3 user-based security model (USM) settings to access devices.

To view the current list of SNMPv3 USM settings for a Region:

1. Navigate to the **SNMPv3 Settings** tab on the Communication Region form.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Select the **Communication Configuration**.
   c. Navigate to the **Regions** tab.
   d. Do one of the following:
      - To create a region definition, click the New icon.
      - To edit a region definition, select a row, click the Open icon.
   e. In the **Communication Region** form, access the **SNMPv3 Settings** tab.
2. The displayed table lists the Unique Name of each SNMPv3 USM setting for this region.

   Note: NNMi tries to use the Specific Node SNMPv3 Settings. If none match, NNMi tries the Region SNMPv3 Settings provided here. If none match, NNMi tries the default SMNPv3 settings.
3. You can also do the following:
   - To establish a new setting, click the New icon. See "Communication Region SNMPv3 Settings form" on the next page.

     Click **Save and Close** to return to the Communication Region form.
   - To edit an existing setting, select a row, click the Open icon. See "Communication

Region SNMPv3 Settings form" on the next page.

Click **Save and Close** to return to the Communication Region form.

- To delete a setting from the Region's list, select a row and click the Delete icon.

  Note:The record remains in the database for possible use elsewhere and is simply removed from this Communication Region's list.

4. Click **Save and Close** to return to the Communication Configuration form.
5. Click **Save and Close** to apply your changes.

Communication Region SNMPv3 Settings form

NNMi can use SNMPv3 user-based security model (USM) settings to access devices.

NNMi tries to use the current SNMPv3 Settings attribute value from Specific Node Settings. If none match, NNMi tries the Region SNMPv3 Settings provided here. If none match, NNMi tries the default SMNPv3 settings.

To configure an SNMPv3 Setting for a Region:

1. Navigate to the **Communication Region SNMPv3 Settings** form.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Select the **Communication Configuration**.
   c. Navigate to the **Regions** tab.
   d. Do one of the following:
      - To create a region definition, click the New icon.
      - To edit a region definition, select a row, click the Open icon.
   e. In the **Communication Region** form, navigate to the **SNMPv3 Settings** tab.
   f. Do one of the following:
      - To create an SNMPv3 Setting definition, click the New icon.
      - To edit an SNMPv3 Setting, select a row, click the Open icon.
      - To remove an SNMPv3 Setting from this Region, select a row, click the Delete icon.

        Note: The record remains in the database for possible use elsewhere and is simply removed from this Communication Region's list.
2. Click the SNMPv3 Settings Lookup icon and select one of the options from the drop-down menu:
   - Show Analysis to display Analysis Pane information for the currently configured (selected) SNMPv3 Setting name. (See Use the Analysis Pane for more information about the Analysis Pane.)

- Quick Find to view and select from the list of all existing SNMPv3 Settings (for more information see "Use the Quick Find Window" on page 41).
- Open to display the details of the currently configured (selected) SNMPv3 Setting (see "SNMPv3 Settings Form" for more information).
- New to create a new SNMPv3 Setting (see "SNMPv3 Settings Form" for more information).

3. Click **Save and Close** to return to the Communication Region SNMPv3 Settings form.
4. Click **Save and Close** to return to the Communication Region form.
5. Click **Save and Close** to return to the Communication Configuration form.
6. Click **Save and Close** to apply your changes.

Configure Credential Settings for Regions

NNMi uses the Device Credentials settings for the following:

- Device discovery of some vendor-specific devices that require non-SNMP communication, such as Netconf over SSH. For a list of the these devices see the NNMi Device Support Matrix.
- *HP Network Node Manager iSPI Network Engineering Toolset Software*

NNMi uses the following sequence to determine Device Credentials:

- Use the Specific Node Device Credentials. If none match, continue.
- Use the Region Device Credentials (provided here). If none match, continue.
- Use the Default Credential settings.

To provide credential settings for this region:

1. Navigate to the **Region Device Credentials** form.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Select **Communication Configuration**.
   c. Navigate to the **Regions** tab.
   d. Do one of the following:
      - To establish a region definition, click the New icon.
      - To edit a region definition, select a row, click the Open icon.
   e. In the **Communication Region** form, navigate to the **Device Credentials** tab.
   f. Do one of the following:
      - To establish a credential setting, click the New icon, and continue.
      - To edit a credential setting, select a row, click the Open icon, and continue.
      - To delete a credential setting, select a row and click the Delete icon.
2. Provide the attribute values of credentials for this region (see table).

3. Click **Save and Close** to return to the Communication Region form.
4. Click **Save and Close** to return to the Communication Configuration form.
5. Click **Save and Close** to apply your changes.

Note: *HP Network Node Manager iSPI Network Engineering Toolset Software* uses the Default Credentials setting to access devices when running Diagnostics either automatically or when the **Actions** → **Run Diagnostics (iSPI NET only)** option is used. (See "Configure Diagnostics for an Incident (NNM iSPI NET)" on page 757 and Node Form: Diagnostics Tab for more information.)

> **Tip:** You can right-click any object in a table or map view to access the **Actions** menu.

At each level in the sequence to determine the Device Credentials (see bullet list above), NNMi first uses Secure Shell (SSH) to establish a secure connection, and if the SSH attempt fails, NNMi tries Telnet protocol as the communication method.

> **Caution:** By default, neither Microsoft Internet Explorer nor Mozilla Firefox defines the telnet command nor the SSH command, so using either of these menu items produces an error message. See the "Configuring the Telnet and SSH Protocols for Use by NNMi" chapter in the *HP Network Node Manager i Software Deployment Reference* for configuration information.

Device Credential Attributes for this Region

Attribute	Description
User Name	Type the user name that you want NNMi to use for logging into devices in this Communication Region.
Password	Type the password that you want NNMi to use for logging into devices in this Communication Region. **Note:** NNMi encrypts the password and displays asterisks for this attribute. If you want to change the password, first clear the asterisks displayed in the Password attribute and enter the new Password value.

Configure Specific Nodes (Communication Settings)

Configuring communication protocols for specific devices is optional.

Use the Specific Node Settings tab to fine tune communication protocol usage and settings for a particular device within your environment. For example, provide settings for your most important devices, or disable communication with the least important devices.

When you leave a field blank, NNMi uses the next applicable configuration setting for that field in the following order:

- The value configured for a Region that includes this device. If multiple Region definitions include this device (for example, buildings, floors within those buildings, or workgroups within a particular floor), the first match applies (the matching region with the lowest Ordering number) . See "Configure Regions (Communication Settings)" on page 136.
- The default value for this field (see "Configure Default SNMP, Management Address, and ICMP Settings" on page 120, "Configure Default Community Strings (SNMPv1 or SNMPv2c)" on page 129, "Configure the Default Device Credentials" on page 135, and "Configure Default SNMPv3 Settings" on page 133.

> **Note:** NNMi enables you to set up one or more SNMP Proxy Servers in the cases where an SNMP node is otherwise unreachable (for example, when a node you want to manage is behind a firewall). To enable NNMi to use the SNMP Proxy Server, when you configure communication protocols for specific devices, you must include the IP address and port number on the SNMP Proxy Server. See "Specific Node Settings Form (Communication Settings)" below for more information.

To configure specific devices, you have two choices:

- "Specific Node Settings Form (Communication Settings)" below.
- "Load Communication Settings from a File" on page 168

Specific Node Settings Form (Communication Settings)

Create specific node settings to control the way NNMi monitors your most important devices or least important devices.

Tip: If no value is provided for an attribute in the Communication Node form, NNMi uses the applicable Region settings and if none match, NNMi uses the default settings.

If configuring Specific Node Settings, also see the *HP Network Node Manager i Software Deployment Reference* which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`

To configure communication protocol settings for a specific node:

1. Access the **Specific Node Settings** form.
    a. From the workspace navigation panel, select the **Configuration** workspace.
    b. Select the **Communication Configuration**.
    c. Navigate to the **Specific Node Settings** tab.
    d. Do one of the following:
        - To establish settings for a node, click the New icon, and continue.
        - To edit settings for a node, select a row, click the Open icons, and continue.

- To delete settings for a node, select a row and click the ✖ Delete icon.

2. Provide the communication protocol settings for the node (see the Basic Settings table, SNMP Settings table, and ICMP Settings table).

   For an explanation of how NNMi implements timeout and retry configurations, see "Timeout / Retry Behavior Example for SNMP" on page 127 and "Timeout / Retry Behavior Example for ICMP" on page 128.

3. *Optional*. Make additional configuration choices. Click here for a list of choices .

4. Click **Save and Close** to return to the Communication Configuration form.

5. Click **Save and Close** to apply your changes.

Basic Settings for this Device

Attribute	Description
Target Hostname	The Hostname attribute value from the Node form of the discovered node must match what is entered here. Case-insensitive, NNMi automatically converts the hostname to all lowercase on the Node form. NNMi follows a set of rules to dynamically generate the value stored in the NNMi database for each Node's Hostname. Click here for details. • If the Node supports SNMP, NNMi requests the Hostname using the IP Address of the associated SNMP agent (the Management Address attribute value on the Node form). When the NNMi administrator chooses **Enable SNMP Address Rediscovery** ☑ in the Communication Configuration: ▪ If the SNMP Agent does not respond, NNMi checks for another Management Address to request the Hostname, and the Hostname could change. ▪ If the SNMP Agent associated with the node changes, the Management Address and Hostname could change. When the NNMi administrator disables **Enable SNMP Address Rediscovery** □ in the Communication Configuration: ▪ If the SNMP Agent does not respond, NNMi uses the previously gathered Management Address attribute value to request the Hostname. ▪ If the SNMP Agent associated with the node changes, NNMi uses the previously gathered Management Address attribute value to request the Hostname. • If the Node does not support SNMP, no Management Address is available. NNMi requests a Hostname starting with the lowest IP Address associated with the node (a Discovery Seed value or an IP address value gathered from a neighboring device). NNMi uses the first Hostname provided. The Hostname might change during a future discovery cycle.

Basic Settings for this Device , continued

Attribute	Description
	Note: NNMi administrators can use NNMi property file settings to change the way NNMi determines Hostname values: • `nms-topology.properties` file settings: If DNS is the source of the Node's Hostname, there are three choices. By default NNMi uses the exact Hostname from your network configuration. It is possible to change NNMi behavior to convert Hostnames to all uppercase or all lowercase. See the "Modifying NNMi Normalization Properties" section of the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`. • `nms-disco.properties` file settings: The Hostname is either requested from the Node's lowest loopback interface IP address that resolves to a Hostname or requested from the Node's designated Management Address (SNMP agent address). With either choice, when no IP address resolves to a Hostname, the IP address itself becomes the Hostname. See the "Maintaining NNMi" chapter of the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.
Preferred Management Address	Do one of the following: • Specify the address you want NNMi to use for SNMP communications with this device. If you enter an invalid or unreachable address, the device is not discovered or monitored. • Leave this attribute empty. NNMi dynamically selects the management address, based on responses from the device's SNMP agent and your choices in "Configure Default SNMP, Management Address, and ICMP Settings" on page 120. **Note**: The NNMi administrator can over-ride this setting. See the Enable SNMP Communication attribute and the Enable SNMP Address Rediscovery attribute settings.
Description	*Optional*. Provide a description for this configuration that would be useful for communication purposes within your team. Type a maximum of 255 characters. Alpha-numeric, spaces, and special characters (~ ! @ # $ % ^ &amp; * ( ) _+) are permitted.

SNMP Settings for this Device

Attribute	Description
Enable SNMP	If ☑ enabled, the Discovery Process and State Poller Service generate

SNMP Settings for this Device , continued

Attribute	Description
Communication	network traffic with SNMP protocol to discover and monitor this device. **Note**: Your choice might be overridden if Monitoring Configuration settings disable SNMP usage for the State Poller Service, see "Global Control Settings for Monitoring" on page 343 or "Configure NNMi Monitoring Behavior" on page 340. If □ disabled, NNMi does not generate any SNMP traffic to this device. **Caution**: With no SNMP data, Spiral Discovery interprets each IP Address as a separate node, Causal Engine calculates Status based only on IP address State, previously discovered Interfaces show a State attribute value of "Not Polled" and a Status attribute value of "No Status" with the Interface map-symbol color set to beige, and no new Interfaces are discovered.
Enable SNMP Address Rediscovery	**Note**: The NNMi administrator can over-ride this setting. See the Enable SNMP Communication and the Preferred Management Address attributes. If ☑ enabled, NNMi automatically identifies which management address (SNMP agent) to use for each device. If the initially configured address becomes unreachable, NNMi automatically locates another address, if possible, and changes the management address attribute value. Click here for more information. When NNMi first discovers a node, the *seed address* (provided by the NNMi administrator) or discovered address (for non-seeded nodes) becomes the initial address used for SNMP communication. After NNMi builds an inventory of all IP addresses associated with the node (see "How Spiral Discovery Works" on page 176), NNMi follows a set of rules to determine which address is the best choice for each node's Management Address: **Note:** With NNMi Advanced, the NNMi administrator specifies whether NNMi prefers IPv4 or IPv6 addresses when selecting the Management Address. See Configure Default SNMP, Management Address, and ICMP Settings. 1. NNMi ignores the following addresses when determining which Management Address is most appropriate: ▪ Any address of an administratively-down interface. ▪ Any address that is virtual (for example, **VRRP**[1]). ▪ Any IPv4 **Anycast Rendezvous Point IP Address**[2] or IPv6 Anycast

[1] Virtual Router Redundancy Protocol
[2] Rendezvous Point addresses are loopback addresses used for routers in multi-cast network configurations.

SNMP Settings for this Device , continued

Attribute	Description
	address. ▪ Any address in the reserved loopback network range. IPv4 uses 127/24 (`127.*.*.*`) and IPv6 uses `::1`. ▪ Any **IPv6 link-local address**[1]. 2. If the NNMi Administrator specifies Enable SNMP Address Rediscovery in Communication Configuration, NNMi prefers the last-known Management Address (if any). 3. If the Management Address does not respond and the NNMi Administrator specifies Enable SNMP Address Rediscovery in Communication Configuration, NNMi uses the Communication Configuration settings for *Management Address Selection*. The NNMi Administrators chooses the order in which NNMi checks the following: ▪ Seed IP / Management IP - If the NNMi Administrator configures a Seed, NNMi uses the Seed address (either a specified IP address or the DNS address associated with a specified hostname) only during initial Discovery. NNMi then requests the current Management Address (the address from which the node's SNMP Agent responds) and uses that IP address for all communication after initial discovery. ▪ Lowest Loopback - If a node supports multiple **loopback address**[2], NNMi queries each loopback addresses, starting with the lowest number. NNMi uses the loopback address with the lowest number from which the SNMP agent responds (for example, 10.16.42.197 is a lower number than 10.16.197.42). ▪ Highest Loopback - If a node supports multiple **loopback address**[3], NNMi queries each loopback addresses, starting with the highest number. NNMi uses the loopback address with the highest number from which the SNMP agent responds.

[1] A non-routable IPv6 unicast address only used for communication with other nodes on the same link (LAN or VLAN). Link local addresses cannot be used for communication that must be forwarded through a router. IPv6 auto-configuration automatically assigns a unique link local address in the fe80::/10 address space to each IPv6-enabled interface on a system.

[2] The address associated with the loopback interface. The loopback interface is a virtual interface on a device that provides a route for internal communication. Many vendors provide a specially configured loopback for management purposes. Exact details of how loopbacks are configured varies by vendor and model. See each device's documentation for details. NNMi identifies these loopback addresses by using ifType Number 24, softwareloopback from the IANA ifType-MIB.

[3] The address associated with the loopback interface. The loopback interface is a virtual interface on a device that provides a route for internal communication. Many vendors provide a specially configured loopback for management purposes. Exact details of how loopbacks are configured varies by vendor and model. See each device's documentation for details. NNMi identifies these loopback addresses by using ifType Number 24, softwareloopback from the IANA ifType-MIB.

SNMP Settings for this Device , continued

Attribute	Description
	- Interface Matching - The NNMi Administrator chooses which interface MIB variable NNMi queries to detect changes. NNMi can use the following MIB-II attribute values: `ifIndex`, `ifName`, `ifDescr`, `ifAlias`, or a combination of these (`ifName` or `ifDescr`, `ifName` or `ifDescr` or `ifAlias`). 4. If no response, NNMi queries any remaining IP addresses in the node's IP address inventory, starting with the lowest number. NNMi uses the address with the lowest number from which the SNMP agent responds. 5. If no response, NNMi checks for any Mapped Address configured for one of the currently known addresses (see the Mapped Address column in the Custom IP Addresses view). **Note:** The address represents a *static* Network Address Translation (NAT) pair's *external IP address* from the internal/external IP address pair. NNMi Administrators configure these pairs using the Overlapping IP Address Mapping form. NNMi uses this list of addresses starting with IPv4 from low to high, then IPv6 from low to high. 6. If no response, NNMi might be configured to repeat the sequence using SNMPv1, SNMPv2c, or SNMPv3 in the order specified by the NNMi administrator (Communication Configurations *SNMP Minimum Security Level* settings). 7. When all else fails, NNMi retains the last known Management Address (if any) and automatically changes the State of that SNMP Agent object to Critical. This process is repeated during each Spiral Discovery cycle, and the Management Address can change. For example, NNMi's inventory of addresses for the node expands, or the current Management Address does not respond to SNMP queries due to network problems or node reconfiguration. The NNMi administrator can prevent changes to the management address using the Communication Configurations *Enable SNMP Address Rediscovery* or *Preferred Management Address* setting. If □ disabled, when the current management address (SNMP agent) becomes unreachable, NNMi reclassifies the node as a non-SNMP node until the previously configured management address is available again.
Enable SNMP GetBulk	*Applies only to SNMPv2 or higher.* If you have devices in your network environment that have trouble responding to `GetBulk` commands, you can instruct NNMi to use `Get` or `GetNext` instead of `GetBulk`. If ☑ enabled, NNMi uses the SNMPv2c `GetBulk` command to gather information from this device. If □ disabled, NNMi uses the SNMP `Get` or `GetNext` command to gather

SNMP Settings for this Device , continued

Attribute	Description
	information from this device (requesting responses for one SNMP OID at a time).
SNMP Timeout	(Seconds:Milliseconds) Maximum 1 millisecond less than a minute: 59 seconds 999 milliseconds. Time that NNMi waits for a response to an SNMP query before reissuing the request. Both the Discovery Process and the State Poller Service use this setting for this device. For an explanation of how NNMi implements timeout and retry configurations, see "Timeout / Retry Behavior Example for SNMP" on page 127.
SNMP Retries Count	Maximum number of retries that NNMi issues for an SNMP query before determining the query result to be "unresponsive". Zero means no retries. Both the Discovery Process and the State Poller Service use this setting for this device.
SNMP Port	Default is 161. Specifies the management server's port that NNMi uses when generating SNMP traffic. Both the Discovery Process and the State Poller Service use this setting for this device.
SNMP Proxy Address	*Optional*. IP address of the your SNMP Proxy Server (for example, a proxy that gathers data from non-SNMP devices and can use that data to respond to NNMi SNMP requests). To enable a proxy, you must also provide the port number of your SNMP Proxy Server. See SNMP Proxy Port (next attribute). **Note**: When you configure NNMi to use a Proxy Server, you must ensure that the Proxy Server vendor supports the Object Identifiers used to handle SNMP requests and responses. See the “NNMi Communications” chapter of the HP Network Node Manager i Software Deployment Reference for more information.
SNMP Proxy Port	*Optional*. Port number of the SNMP Proxy Server. To enable a proxy, you must also provide the IP address of your SNMP Proxy Server. See SNMP Proxy Address (previous attribute). **Note**: When you configure NNMi to use a Proxy Server, you must ensure that the Proxy Server vendor supports the Object Identifiers used to handle SNMP requests and responses. See the “NNMi Communications” chapter of the HP Network Node Manager i Software Deployment Reference for more information.
SNMP Preferred Version	This setting determines whether each NNMi Rediscovery cycle automatically detects the best SNMP choice (v1, v2, or v3) for this Node (automatically detects any upgrade to the SNMP agent on each Node), or uses only the SNMP version that you specify. Specifies the SNMP version that NNMi should use when communicating with

SNMP Settings for this Device , continued

<table>
<tr><th>Attribute</th><th>Description</th></tr>
<tr><td></td><td>a device. Select one of the following options:
<table>
<tr><td>1</td><td>Indicates you want NNMi to try only SNMPv1 settings.
<b>Tip</b>: Use this option when you do not want NNMi to use <code>GetBulk</code> commands on the device.</td></tr>
<tr><td>2</td><td>Indicates you want NNMi to use SNMPv2c settings, and, if that fails, try SNMPv1 settings. If both SNMPv2c and SNMPv1 fail, NNMi tries SNMPv3 settings if any are available.</td></tr>
<tr><td>3</td><td>Indicates you want NNMi to use SNMPv3 settings for this device. NNMi uses the <b>SNMPv3 Settings</b> configuration to determine which of the following User-based Security Module (USM) levels of security to provide:
• No Authentication, No Privacy
• Authentication, No Privacy
• Authentication, Privacy
See "Configure Default SNMP, Management Address, and ICMP Settings" on page 120 for more information.</td></tr>
</table>
</td></tr>
</table>

Note: The SNMP Minimum Security Level is determined by the settings on the Communication Configurations' Specific Node Settings form, **SNMPv3 Settings** tab where SNMPv3 Settings for this Node are established.

ICMP Settings for this Device

Attribute	Description
Enable ICMP Communication	If ☑ enabled, NNMi generates network traffic with ICMP protocol to this device. If □ disabled, NNMi does not generate any ICMP traffic to this device: • Addresses in this Node (both previously discovered and newly discovered) have a State attribute value of "Not Polled" and a Status attribute value of "No Status" with the IP Address map-symbol color set to beige. • If both ICMP and SNMP are disabled, the Node has a Status attribute value of "No Status" have a map-symbol background shape color set to beige. **Note**: Your choice might be overridden if Monitoring Configuration settings disable ICMP usage for the State Poller Service, see "Global Control Settings for Monitoring" on page 343 or "Configure NNMi Monitoring Behavior" on page 340.

ICMP Settings for this Device , continued

Attribute	Description
ICMP Timeout	(Seconds:Milliseconds) Maximum 1 millisecond less than a minute: 59 seconds 999 milliseconds. Time that NNMi waits for a response to an ICMP query before reissuing the request to this device. For an explanation of how NNMi implements timeout and retry configurations, see "Timeout / Retry Behavior Example for ICMP" on page 128.
ICMP Retries Count	Maximum number of retries that NNMi issues for an ICMP query to this device before logging an error. Zero means no retries.

Related Topics:

"Configure Default SNMP, Management Address, and ICMP Settings" on page 120

"Configure Default Community Strings (SNMPv1 or SNMPv2c)" on page 129

"Configure Regions (Communication Settings)" on page 136

Configure SNMPv1/v2c Community Strings for a Specific Node

Optional. Configure the SNMPv1 or SNMPv2c community strings for each node.

NNMi uses the SNMPv2c settings to discover the SNMPv2c information about your network. This also determines whether NNMi *receives or discards incoming* SNMPv2c traps. Click here for more information.

- If the *incoming* trap's Source Node (and sometimes Source Object, such as card or interface) has not yet been discovered by NNMi, NNMi discards the trap. See "Handle Unresolved Incoming Traps" on page 776 for additional information. See also "Configure Network Devices to Send SNMP Notifications to NNMi" on page 771.

- If the Source Node was not discovered using SNMv3, NNMi discards any incoming SNMPv3 traps from that Node.

- NNMi discards traps that have no incident configuration or with an incident configuration set to Disabled. To ensure that NNMi retains all received Trap instances when your network environment includes SNMP agents using a variety of SNMPv1, SNMPv2c, and SNMPv3 protocol, you must configure two Incidents: one for the SNMPv1 version and one for the SNMPv2c/3 version of that trap. See "Configure SNMP Trap Incidents" on page 782.

- If either the Source Node or Source Object has *Management Mode* set to **Not Managed** or **Out of Service** in the NNMi database, NNMi always discards the incoming trap. See "Understand the Effects of Setting the Management Mode to Not Managed or Out of Service " on page 464.

  NNMi provides the Management Mode workspace so that you can quickly view lists of all nodes, interfaces, cards, addresses, or node components that NNMi is not currently discovering or monitoring. For information about these views:

- NNMi discards most incoming traps from network objects that are not monitored. For example, you can configure NNMi to exclude specified interfaces from being monitored. See "Monitoring Network Health" on page 340 for more information.

Note: If you want the NNMi management server to *forward* SNMPv2c traps to other machines in your network environment, see "Configure Trap Forwarding" on page 1376 for additional configuration steps.

To provide SNMPv1/v2c community strings for a specific device:

1. Navigate to the **Specific Node Settings** form.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Select **Communication Configuration**.
   c. Navigate to the **Specific Node Settings** tab.
   d. Do one of the following:
      - To establish a node definition, click the New icon.
      - To edit a node definition, select a row, click the Open icon.
2. Navigate to the **SNMPv1/v2c Community Strings** tab.
3. To provide a *read community string*, navigate to the **Read Community String** attribute and provide the appropriate string (see table).

   Tip: If you do not provide any read community string, NNMi uses the applicable Region settings and if none match, NNMi uses the default settings .
4. To provide a *write community string*, navigate to **the Write Community String** attribute and provide the appropriate string (see table).

   Tip: If you do not provide any write community string, NNMi uses the applicable Region setting and if none match, NNMi uses the default setting .
5. Click **Save and Close** to return to the Communication Configuration form.
6. Click **Save and Close** to apply your changes.

SNMPv1 or SNMPv2c Community String for this Device

Attribute	Description
Read Community String	The SNMPv1 or SNMPv2c "Get" (read-only) Community String that is used for this device (case-sensitive). **Tip**: If you do not provide any read community string, NNMi uses the applicable Region settings and if none match, NNMi uses the default settings . Many proxy vendors use the *read community string* for specifying remote target information. NNMi supports substitution parameters within read community strings for SNMPv1 or SNMPv2c proxy environments. Click here for more information. Copy and paste these codes at the end of your read community string to provide the values required by your proxy environment. NNMi substitutes the actual attribute values from the NNMi database at runtime:

SNMPv1 or SNMPv2c Community String for this Device , continued

Attribute	Description
	${contextName} = Used for specifying VLAN context for switches (VLAN associated with the remote target node) ${managementAddress} = Node form, Management Address attribute value (the remote target node) ${snmpPort} = SNMP Agent form, UDP Port attribute value (SNMP agent associated with the remote target node) Type a maximum of 255 characters. Alpha-numeric, spaces, and special characters (~ ! @ # $ % ^ &amp; * ( ) _+ -) are permitted.
Write Community String	*Optional*. For use with the nnmsnmpset.ovpl command line tool. The SNMPv1 or SNMPv2c "Set" (write) Community String that is used for the SNMP Agent for each node specified (case-sensitive). **Tip**: SNMP Agents are often configured with different community strings for "Set" requests than for "Get" (read) requests. SNMPv1 and SNMPv2c require that you know the SNMP agent's *write community string* before you can change settings on any device. The nnmsnmpset.ovpl command can use the value you provide here, rather than requiring that you type the write community string each time you invoke the command. **Tip**: If you do not provide any write community string, NNMi uses the applicable Region setting and if none match, NNMi uses the default setting. Type a maximum of 255 characters. Alpha-numeric, spaces, and special characters (~ ! @ # $ % ^ &amp; * ( ) _+ -) are permitted. Because this is a type of password, you must enter the value twice.

Configure SNMPv3 Settings for a Specific Node

NNMi can use SNMPv3 user-based security model (USM) settings to access devices.

NNMi uses the current SNMPv3 Settings provided for a node, if available.

To configure an SNMPv3 Settings for a specific node:

1. Navigate to the **Specific Node Settings** form.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Select **Communication Configuration**.
   c. Navigate to the **Specific Node Settings** tab.
   d. Do one of the following:
      - To establish a node definition, click the New icon, and continue.
      - To edit a node definition, select a row, click the Open icon, and continue.

2. Navigate to the **SNMPv3 Settings** tab.
3. Click the SNMPv3 Settings Lookup icon and select one of the options from the drop-down menu:
   - Show Analysis to display Analysis Pane information for the currently configured (selected) SNMPv3 Setting name. (See Use the Analysis Pane for more information about the Analysis Pane.)
   - Quick Find to view and select from the list of all existing SNMPv3 Settings (for more information see "Use the Quick Find Window" on page 41).
   - Open to display the details of the currently configured (selected) SNMPv3 Setting (see "SNMPv3 Settings Form" for more information).
   - New to create a new SNMPv3 Setting (see "SNMPv3 Settings Form" for more information).
4. Click **Save and Close** to return to the Specific Node Settings form.
5. Click **Save and Close** to return to the Communication Configuration form.
6. Click **Save and Close** to apply your changes.

Configure Credential Settings for a Specific Node

NNMi uses the Device Credentials settings for the following:

- Device discovery of some vendor-specific devices that require non-SNMP communication, such as Netconf over SSH. For a list of the these devices see the NNMi Device Support Matrix.
- *HP Network Node Manager iSPI Network Engineering Toolset Software*

NNMi uses the following sequence to determine Device Credentials:

- Use the Specific Node Device Credentials (provided here). If none match, continue.
- Use the Region Device Credentials. If none match, continue.
- Use the Default Credential settings.

To provide credential settings for a specific node:

1. Navigate to the **Specific Node Device Credentials** form.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Select **Communication Configuration**.
   c. Navigate to the **Specific Node Settings** tab.
   d. Do one of the following:
      - To establish a definition, click the New icon.
      - To edit a definition, click the Open icon in the row representing the configuration you want to edit.
   e. In the **Specific Nodes Settings** form, navigate to the **Device Credentials** tab.

f. Do one of the following:

  - To establish a credential setting, click the ✳ New icon, and continue.
  - To edit a credential setting, click the Open icon in the row representing the configuration you want to edit, and continue.
  - To delete a credential setting, select a row and click the ✖ Delete icon

2. Provide the attribute values of credentials for this node (see table).

   Note: NNMi tries to use the Specific Node Device Credentials provided here. If none match, NNMi tries the Region Device Credential settings. If none match, NNMi tries the Default Device Credentials.

3. Click **Save and Close** to return to the Specific Node Settings form.
4. Click **Save and Close** to return to the Communication Configuration form.
5. Click **Save and Close** to apply your changes.

Note: *HP Network Node Manager iSPI Network Engineering Toolset Software* uses the Default Credentials setting to access devices when running Diagnostics either automatically or when the **Actions** → **Run Diagnostics (iSPI NET only)** option is used. (See "Configure Diagnostics for an Incident (NNM iSPI NET)" on page 757 and Node Form: Diagnostics Tab for more information.)

Tip: You can right-click any object in a table or map view to access the **Actions** menu.

At each level in the sequence to determine the Device Credentials (see bullet list above), NNMi first uses Secure Shell (SSH) to establish a secure connection, and if the SSH attempt fails, NNMi tries Telnet protocol as the communication method.

Caution: By default, neither Microsoft Internet Explorer nor Mozilla Firefox defines the telnet command nor the SSH command, so using either of these menu items produces an error message. See the "Configuring the Telnet and SSH Protocols for Use by NNMi" chapter in the *HP Network Node Manager i Software Deployment Reference* for configuration information.

Specific Node Device Credential Attributes

Attribute	Description
User Name	Type the user name that you want NNMi to use for logging into this device.
Password	Type the password that you want NNMi to use for logging into this device. **Note:** NNMi encrypts the password and displays asterisks for this attribute. If you want to change the password, first clear the asterisks displayed in the Password attribute and enter the new Password value.

Load Communication Settings from a File

NNMi enables you to import the SNMPv1 or SNMPv2c community strings (read and write) or the SNMPv3 USM settings for either specifc nodes or comunication regions. This is useful when your SNMP is managed by a change control mechanism. You can bulk insert the SNMP assignments into NNMi.

Each assignment shows up as an individual entry in one of the following tables on the **Communication Configuration** form:

- **Specific Node Settings** tab
- **Communication Regions** tab

If configuring Specific Node Settings, for additional information, see the *HP Network Node Manager i Software Deployment Reference* which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`. If configuring Communication Regions, see "Configure Regions (Communication Settings)" on page 136 for more information.

To import communication settings:

1. On the NNMi management server's hard drive, create a text file according to the specifications in the nnmcommload.ovpl reference page. Create one line for each device. For more information, see nnmcommload.ovpl

   To add comments to your file, place a # character at the beginning of each comment line.

   Note: When you load this file, the data in the file overwrites any previously entered information about each Hostname (*case-sensitive*).

2. Use the following command line command to load the information into the NNMi database:

   If you do not want to enter an NNMi User Name attribute value and an NNMi Password attribute value at the command line, you can use the nnmsetcmduserpw.ovpl command to specify the valid user name and password (instead of `-u` and `-p`). The credentials set using the `nnmsetcmduserpw.ovpl` command are valid for command execution by the same user. See "Set Up Command Line Access to NNMi" on page 577 for more information.

   Windows:
   ```
   %NnmInstallDir%\bin\nnmcommload.ovpl -u <NNMiadminUserName> -p
   <NNMiadminPassword> -file <path/filename>
   ```

   UNIX:
   ```
   /opt/OV/bin/nnmcommload.ovpl -u <NNMiadminUserName> -p
   <NNMiadminPassword> -file <path/filename>
   ```

3. Verify that the import worked properly:

   a. From the workspace navigation panel, select the **Configuration** workspace.

   b. Select **Communication Configuration**.

   c. Access the **Specific Node Settings** or **Communication Regions** tab.

4. Review each entry in the table to verify that the import was successful.

To verify the communication configuration for an IP Address, at the command line, type:

Note: For more information, see nnmcommconf.ovpl

If you do not want to enter an NNMi User Name attribute value and an NNMi Password attribute value at the command line, you can use the nnmsetcmduserpw.ovpl command to specify the valid user name and password (instead of `-u` and `-p`). The credentials set using the `nnmsetcmduserpw.ovpl` command are valid for command execution by the same user. See "Set Up Command Line Access to NNMi" on page 577 for more information.

Windows:

```
%NnmInstallDir%\bin\nnmcommconf.ovpl -u <NNMiadminUsername> -p
<NNMiadminPassword> -proto snmp -host <node IP address>
```

UNIX:

```
/opt/OV/bin/nnmcommconf.ovpl -u <NNMiadminUsername> -p
<NNMiadminPassword> -proto snmp -host <node IP address>
```

To verify the ICMP configuration for an IP Address, at the command line, type:

Note: For more information, see nnmcommconf.ovpl

If you do not want to enter an NNMi User Name attribute value and an NNMi Password attribute value at the command line, you can use the nnmsetcmduserpw.ovpl command to specify the valid user name and password (instead of `-u` and `-p`). The credentials set using the `nnmsetcmduserpw.ovpl` command are valid for command execution by the same user. See "Set Up Command Line Access to NNMi" on page 577 for more information.

Windows:

```
%NnmInstallDir%\bin\nnmcommconf.ovpl -u <NNMiadminUsername> -p
<NNMiadminPassword> -proto icmp -host <node IP address>
```

UNIX:

```
/opt/OV/bin/nnmcommconf.ovpl -u <NNMiadminUsername> -p
<NNMiadminPassword> -proto icmp -host <node IP address>
```

Troubleshooting Communication Settings

After you configure your communication settings and wait until Auto-Discovery completes at least one cycle, you can verify your Communication Settings:

- "Verify That All Nodes Support SNMP" on the next page
- "Verify a Node's Communication Settings" on the next page
- "Verify Communication Settings" on page 172
- "Resolve Authentication Errors" on page 173

You can fine tune NNMi's SNMP/ICMP traffic in the following ways:

- Minimize timeouts and retries.

  When NNMi attempts to contact a node using ICMP / SNMP during an Auto-Discovery cycle, the Communication Configuration settings determine what information NNMi can gather. If the correct ICMP / SNMP settings are not provided or if NNMi discovers non-SNMP devices (see "Verify That All Nodes Support SNMP" on the next page), NNMi resorts to timeouts and retries.

Large timeout values or a high number of retries can degrade overall performance of discovery. If your network environment contains nodes that you know respond slowly to ICMP / SNMP requests, consider using the Regions or Specific Nodes settings to fine tune the number of timeouts and retries NNMi uses during each Auto-Discovery cycle.

- Limit the number of *default* SNMPv1/SNMPv2c Community Strings to ensure efficient Auto-Discovery performance. See "Configure Default Community Strings (SNMPv1 or SNMPv2c)" on page 129.
- Limit the number of *default* SNMPv3 user-based security model (USM) settings to ensure efficient Auto-Discovery performance. See "Configure Default SNMPv3 Settings" on page 133.

Verify That All Nodes Support SNMP

After you configure your communication settings and wait until Auto-Discovery completes at least one cycle, check for any nodes that do not respond to SNMP:

1. From the workspace navigation panel, select the **Inventory** workspace.
2. Select the **Nodes** view.
3. Right-click the **Device Profile** column, and select **Create Filter**.
4. Select "contains", and type the following text into **Enter a string**: `No SNMP`.
5. NNMi displays a list of all nodes in your network environment that did not respond to SNMP during Auto-Discovery.
6. Verify that the resulting list is valid.
7. To troubleshoot unexpected results, see:
   - "Verify a Node's Communication Settings" below
   - "Verify Communication Settings" on page 172
   - "Resolve Authentication Errors" on page 173

Verify a Node's Communication Settings

After you configure your communication settings and wait until Auto-Discovery completes at least one cycle, you can check to determine what settings NNMi is using to communicate with a node of interest.

NNMi provides a report about the communication configuration information for a selected node, including the SNMP and ICMP configuration information.

To display a report of a node's current communication settings:

Note: The User Account must be assigned to the **NNMi Administrators** User Group to use this action.

1. Do one of the following:

   Navigate to a table view and select a node:

   a. From the workspace navigation panel, select the workspace of interest. For example,

   Inventory.

   b. Select the view that contains the node with communication settings you want to check. For example, **Nodes**.

   c. Select the row representing the node with communication settings you want to check.

   Navigate to a map view and select a node:

   a. From the workspace navigation panel, select the workspace of interest; for example, **Topology Maps**.

   b. Click the view that contains the node with communication settings you want to check; for example **Initial Discovery Progress** or **Network Overview** map.

   c. From the map view, click the node with communication settings you want to check.

   Navigate to a Node form:

   - From a table view, double-click the row representing the node of interest.
   - From a map view, click the node of interest on the map and click the Open icon.

2. Select **Actions** → **Polling** → **Communication Settings**.

Tip: You can right-click any object in a table or map view to access the **Actions** menu.

Sometimes a device is temporarily not responding properly to SNMP during NNMi's initial discovery, so NNMi makes the wrong decision about which version of SNMP to use. Or perhaps you deployed upgrades to the SNMP agents in your network environment.

To update NNMi's choice of SNMP version used for a Node or Nodes:

Note: The User Account must be assigned to the **NNMi Administrators** User Group to use this action.

1. From the workspace navigation panel, select **Inventory**.
2. Select the **Nodes (All Attributes)** view.
3. Click the **Protocol Version** column heading to sort the view according to SNMP version currently being used by NNMi for communications with each SNMP agent in your network environment.
4. Select all rows that you want NNMi to check for SNMP upgrades or changes.
5. select **Actions** → **Polling** → **Configuration Poll**.

   NNMi reconfigures the SNMP Communication settings by verifying the highest SNMP version available to the SNMP Agent assigned to the node (according to your Communication Configuration settings).

6. Click the **Protocol Version** column heading to resort the view according to SNMP version.
7. Verify that NNMi made the expected changes.

   If still receiving unexpected results, see "Verify Communication Settings" on the next page.

See "Configuring Communication Protocol" on page 119 for information about configuring communication settings.

Related Topics

nnmcommconf.ovpl

Verify Communication Settings

To verify your Communication Configuration settings:

Note: The User Account must be assigned to the **NNMi Administrators** User Group to use this action.

1. Do one of the following:

   Navigate to a table view and select a node:

   a. From the workspace navigation panel, select the workspace of interest. For example, **Inventory**.

   b. Select the view that contains the node with communication settings you want to check. For example, **Nodes**.

   c. Select the row representing the node with communication settings you want to check.

   Navigate to a map view and select a node:

   a. From the workspace navigation panel, select the workspace of interest; for example, **Topology Maps**.

   b. Click the view that contains the node with communication settings you want to check; for example **Initial Discovery Progress** or **Network Overview** map.

   c. From the map view, click the node with communication settings you want to check.

   Navigate to a Node form:

   - From a table view, select the row representing the node of interest.
   - From a map view, click the node of interest on the map and click the Open icon.

2. Select **Actions** → **Configuration Details** → **Communication Settings**.

   Tip: You can right-click any object in a table or map view to access the **Actions** menu.

   NNMi displays a report showing ICMP and SNMP communication configuration settings for this node's SNMP Agent.

   (*NNMi Advanced*) If the Global Network Management feature is enabled and you are signed into a Global Manager:

   - Node managed by the Global Manager = **Actions** → **Configuration Details** → **Communication Settings** opens a report, provided by the Global Manager (NNMi management server).
   - Node managed by a Regional Manager = **Actions** → **Configuration Details** → **Communication Settings** accesses that Regional Manager (NNMi management server) and requests the report.

> **Note:** You must sign into that Regional Manager unless your network environment enables Single Sign-On (SSO) to that Regional Manager through the Global Manager. For more information, see the “Configuring Single Sign-On for Global Network Management” section in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.

Resolve Authentication Errors

To create a list of authentication errors:

1. From the workspace navigation panel, select the **Incident Browsing** workspace.
2. Select an Incident view.
3. Right-click the **Category** column, and select **Create Filter**.
4. Select "equals", and select **Security**.
5. NNMi displays a list of all incidents related to authentication errors; for example an SNMP authentication failure (see also Node Down).

If NNMi generates incidents related to *authentication failure* during discovery, there are several configuration settings that influence authentication errors:

- Communication Configuration.

  Each Node’s Management Address is the address NNMi uses to communicate with the Node’s SNMP agent. The NNMi administrator can control NNMi behavior:

  - Specify the Management Address for a node (in the Communications Configuration, Specific Nodes settings).
  - Otherwise, let NNMi choose an address from all IP addresses associated with each node. This NNMi behavior can be fine-tuned by the NNMi administrator in the Discovery configuration settings.

  Consider configuring smaller Regions with more focused lists of possible access credentials. Or configure Specific Nodes to avoid requiring NNMi to try multiple possible settings.

- Discovery Configuration.

  The following Discovery Configuration fields influence NNMi’s use of SNMP (see "Configure Basic Settings for the Auto-Discovery Rule" on page 218):

  - **Discover Any SNMP Device** field.

    If □ disabled, NNMi discovers only Routers and Switches that respond to SNMP.

    If ☑ enabled, NNMi discovers all devices that respond to SNMP.

  - **Discover Non-SNMP Devices** field.

    If □ disabled, when there is no SNMP response from the device, NNMi does not discover information about the device or add a record of that device to the NNMi database.

    If ☑ enabled,NNMi discovers devices that do not respond to SNMP and assigns the Device Profile named No SNMP as the basis of the database record.

  NNMi's access to SNMP agents is also influenced by the set of rules for choosing management addresses and settings to exclude certain addresses.

- Device Profiles.

  The Device Profiles' **Force Device** attribute setting influences NNMi’s use of SNMP (see Device Profile form).

- Monitoring Configuration.

  NNMi discovers and monitors devices in an ongoing basis (see "Monitoring Network Health" on page 340). For example, when previously discovered SNMP agents quit responding (such as when you reconfigure the device’s SNMP agent), NNMi detects the alternatives.

  To control management address rediscovery after the first NNMi discovery cycle, use Communication Configuration's **Enable SNMP Address Rediscovery** field:

  - If □ disabled, NNMi reports the device as Node Down and does not attempt to find another Communication Configuration setting that works.

  - If ☑ enabled, NNMi retries any configured values in search of one that works.

Chapter 7

Discovering Your Network

Using a wide range of protocols and techniques, NNMi Spiral Discovery gathers a wealth of information about your network inventory, ascertains the relationships between devices (such as subnets and VLANs), and accurately maps out the connectivity between those devices. The NNMi Causal Engine determines the current status of each device (plus each associated interface and address within that device) and proactively notifies you when NNMi detects any trouble or potential trouble.

This dynamic discovery process continues over time. When things change in your network management domain, Spiral Discovery automatically updates information according to a schedule that you set. The topology maps always reflect accurate and timely information about any changes within your network. For more information, see "How Spiral Discovery Works" on the next page.

The first step is to verify that your network environment supports NNMi's Discovery process: "Prerequisites for Discovery" on page 187.

Then establish the Spiral Discovery default settings: "Establish Global Defaults for Spiral Discovery" on page 201 and "Configure Schedule Settings" on page 209

If your network environment includes areas that use network address translation protocols, NNMi can successfully co-exist with the following protocol types (see "Overlapping Address Mapping" on page 191):

- *Static* Network Address Translation (NAT)
- *Dynamic* Network Address Translation (NAT)
- *Dynamic* Port Address Translation (PAT/NAPT)

If your network environment includes areas with conflicting subnet configurations, NNMi can successfully apply subnet masks *separately* to each group of Nodes you identify with a Tenant configuration (see "Configure Tenants" on page 194).

Tip: NNMi's Tenant configuration settings are useful for a variety of situations. Review the Tenant information so you know about all your options.

The NNMi administrator is responsible for the following:

- Decide which nodes NNMi discovers and how often NNMi checks for new devices in your network (see "Configure Discovery " on page 199).
- Specify which devices are the best source of information about your network (see "Specify Discovery Seeds" on page 256).
- Verify that NNMi has an accurate and complete understanding of your network environment (see "Examine Discovery Results" on page 268).
- Change the Discovery configuration as needed over time (see "Keep Your Topology Accurate" on page 277).

Related Topics:

For a list of the types of things NNMi can discover, see About Map Symbols.

From the information collected, NNMi constructs a model of your network configuration in the database, and displays this information in the map views. See View Maps of Network Connectivity for more information about the available map views.

How Spiral Discovery Works

For details about how Spiral Discovery works, see the following:

To ensure that NNMi successfully discovers Nodes in your network environment, verify the following:

- Communication Configuration settings permit NNMi to communicate with all important Nodes using SNMP, ICMP, or both. See "Configuring Communication Protocol" on page 119.
- Prerequisites are met for well-configured SNMP, DNS, and IP address configuration. See "Prerequisites for Discovery" on page 187.
- Global Default settings reflect reality for your network environment. See "Establish Global Defaults for Spiral Discovery" on page 201.

Which Nodes Are Discovered?

You have total control over which Nodes are discovered by configuring a Discovery Seed for each Node. To define a Discovery Seed, you provide one of the following sets of information:

- hostname (*not case-sensitive*) and Tenant
- IP address and Tenant

NNMi uses the Discovery Seed to make initial contact. Discovery seeds are only relevant during initial discovery. NNMi requests each Node's current Management Address (the address from which the node's SNMP Agent responds) and uses that IP address for all communication after initial discovery. You can configure discovery seeds three ways, see "Specify Discovery Seeds" on page 256.

If you choose to use Auto-Discovery, NNMi automatically gathers Hints from each discovered Node and uses that information to find any neighboring devices within your Default Tenant's address range. You simply configure one or both of the following:

- Provide a Discovery Seed for one or more devices
- Enable Ping Sweep and let Auto-Discovery find every device that responds

Discovery seeds are required if any of the following are true:

- You want NNMi to discover only what you specify.
- You want to use discovery seeds as starting points for Auto-Discovery Rules. See "Configure Auto-Discovery Rules" on page 215.
- Your network includes nodes with addresses provided by any of the following protocols (see "Overlapping Addresses in NAT Environments" on page 89):
  - *Static* Network Address Translation (NAT)
  - *Dynamic* Network Address Translation (NAT)
  - *Dynamic* Port Address Translation (PAT/NAPT)
- You want to control which Nodes each NNMi user sees. See "Tenant and Initial Discovery Security Group Assignments" on page 198.

For details about how Spiral Discovery works:

What Information Is Collected?

For details about how Spiral Discovery gathers information, see the following:

NNMi displays the real-time accumulation of information about each Node as it is collected, rather than waiting until Spiral Discovery scans your entire network environment. Spiral Discovery uses a variety of network protocols (read-only queries) within your defined network management domain to gather information about each discovered Node and that Node's connections to other Nodes (see diagram):

1. Information about the node.

   NNMi gathers detailed information about each device. You can review this data on the device's Node form. Examples of configuration details include Tenant, IP address, subnet information, system object ID (RFC 1213, MIB-II `sysObjectID`), number of interfaces, and version of SNMP supported.

   > **Tip:** NNMi does not collect data from Dynamic Host Configuration Protocol (DHCP). Instead, NNMi uses the Media Access Control address (MAC address) of the Node's interfaces to determine a positive ID when hostname changes. See *HP Network Node Manager i Software Deployment Reference* for more information (see **Help** → **Documentation Library**).

2. Conectivity details.

NNMi gathers information about how devices are connected to each other on **Layer 2**[1] and **Layer 3**[2] of your network.

Devices that belong to the Default Tenant can have Layer 2 Connections to any device in any Tenant. Devices within any Tenant *other than* Default Tenant can have Layer 2 Connections *only* to devices within the same Tenant or the Default Tenant.

> **Tip:** NNMi's Tenant configuration settings are useful for a variety of situations. Review the Tenant information so you know about all your options. See "Configure Tenants" on page 194.

During discovery, NNMi reads the Forwarding Database (FDB) tables from Ethernet switches within a network to help NNMi determine communication paths between network devices. NNMi searches these FDB tables for information about discovered nodes. When an NNMi management server finds FDB references to duplicate **MAC addresses**[3]:

- If two or more discovered nodes contain an interface associated with the same Media Access Control (MAC) address within the same Tenant or with one of those nodes in Default Tenant and one in any other Tenant, NNMi disregards the communication paths reported for those duplicate MAC addresses in the FDB. This might result in missing connections on NNMi maps in network areas that include those duplicate MAC addresses.

  (*NNMi Advanced - Global Network Management feature*) If two NNMi management servers discover nodes that contain an interface associated with the same Media Access Control (MAC) address, the Global NNMi management server's maps could be missing connections that are visible on the Regional NNMi management server's maps.

- If a single node contains multiple interfaces that have the same MAC address, NNMi gathers all communication path information for those interfaces and displays that information on NNMi maps.

Forwarding Database (FDB) information can cause NNMi to establish wrong Layer 2 Connections in the following cases:

- When the FDB is configured as cache and contains obsolete data.
- In network environments with hardware from a variety of vendors, when each vendor generates different and sometimes conflicting FDB data.

[1] Refers to the Data Link layer of the multilayered communication model, Open Systems Interconnection (OSI). The Data Link layer moves data across the physical links in the network. The switches and switch-routers are devices that redirect data messages at the layer 2 level, using the destination Media Access Control (MAC) address to determine where to direct the message.

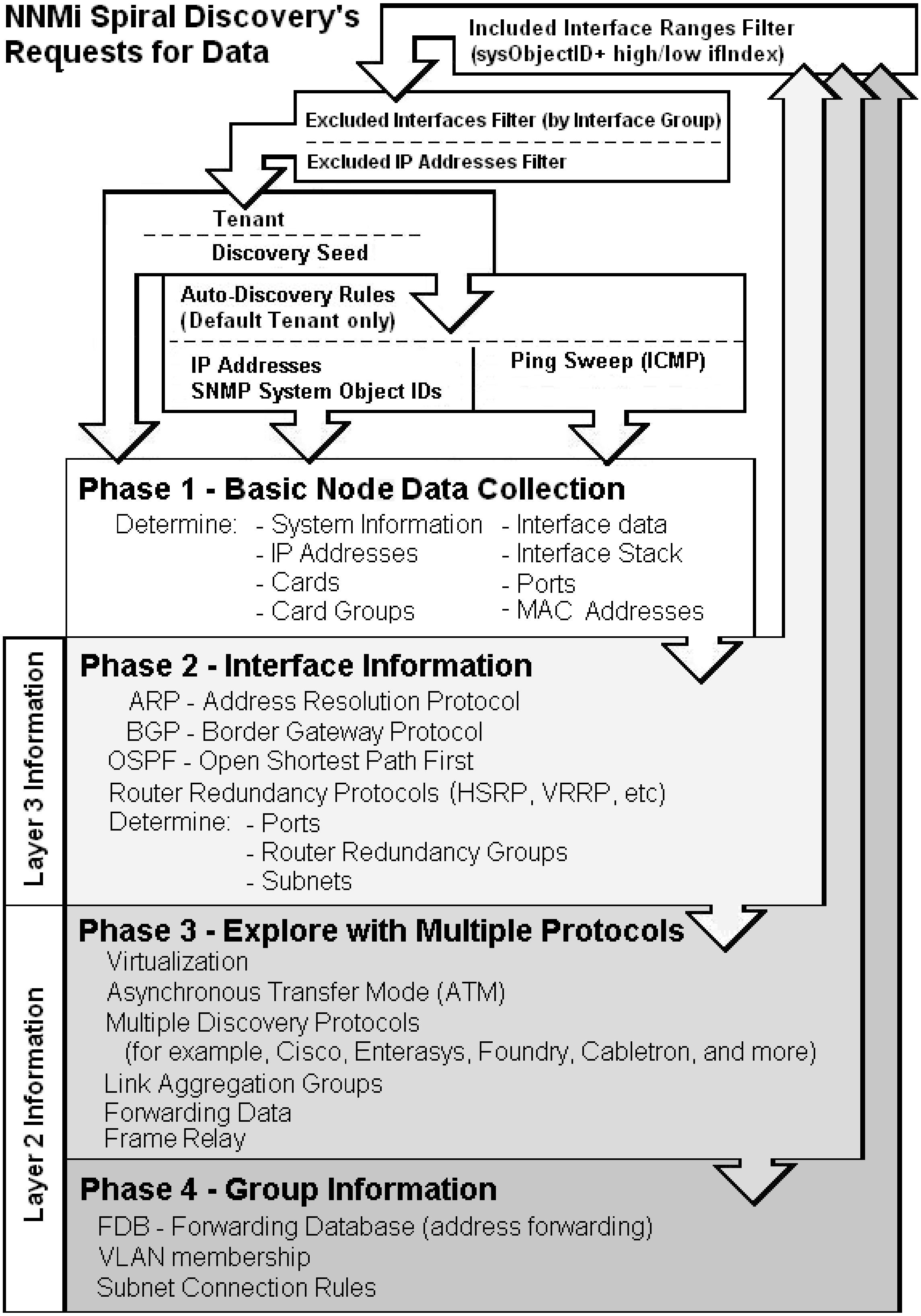
[2] Refers to the Network layer of the multilayered communication model, Open Systems Interconnection (OSI). The Network layer is concerned with knowing the address of the neighboring nodes in the network, selecting routes and quality of service, and recognizing and forwarding incoming messages to local host domains. The router and switch-router are the devices that redirect data messages at the Layer 3 level. Everything in a subnet is connected at the Layer 3 (IP) level.

[3] The Media Access Control address (hardware address or physical address) that the factory burns into a network adapter or device with built-in networking capability. A MAC address has six pairs of hexadecimal digits, separated by colons or dashes. For example 02:1F:33:16:BC:55

Optional: NNMi administrators can configure Spiral Discovery to ignore the FDB data from one Node Group when calculating Layer 2 Connections (the FDB data is still included in other calculations).

For more information, see "Ignore Forwarding Database Information from a Node Group" on page 208.

To create connections that NNMi cannot detect, use IPv4 Subnet Connection Rules. See "Consider IPv4 Subnet Connection Rules" on page 181.


NNMi Spiral Discovery's Requests for Data
Included Interface Ranges Filter (sysObjectID+ high/low ifIndex)
Excluded Interfaces Filter (by Interface Group)
Excluded IP Addresses Filter
Tenant
Discovery Seed
Auto-Discovery Rules (Default Tenant only)
IP Addresses
SNMP System Object IDs
Ping Sweep (ICMP)
Phase 1 - Basic Node Data Collection
Determine: - System Information - IP Addresses - Cards - Card Groups - Interface data - Interface Stack - Ports - MAC Addresses
Phase 2 - Interface Information
ARP - Address Resolution Protocol
BGP - Border Gateway Protocol
OSPF - Open Shortest Path First
Router Redundancy Protocols (HSRP, VRRP, etc)
Determine: - Ports - Router Redundancy Groups - Subnets
Phase 3 - Explore with Multiple Protocols
Virtualization
Asynchronous Transfer Mode (ATM)
Multiple Discovery Protocols (for example, Cisco, Enterasys, Foundry, Cabletron, and more)
Link Aggregation Groups
Forwarding Data
Frame Relay
Phase 4 - Group Information
FDB - Forwarding Database (address forwarding)
VLAN membership
Subnet Connection Rules
Layer 3 Information
Layer 2 Information

For details about how Spiral Discovery works:

Consider IPv4 Subnet Connection Rules

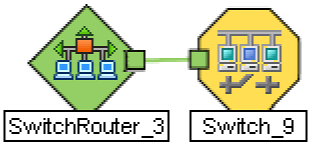
Sometimes it is useful to monitor Layer 2 Connections in the following categories:

- Point-to-point or point-to-multipoint connections between interfaces.
- Virtual IPv4 tunnel connections within your management domain.
- Connections to remote sites (across a Service Provider's network or a WAN).
- Connections among Provider Edge (**PE**[1]) devices in the Default Tenant and Customer Edge (**CE**[2]) devices in Tenants defined by the NNMi administrator.

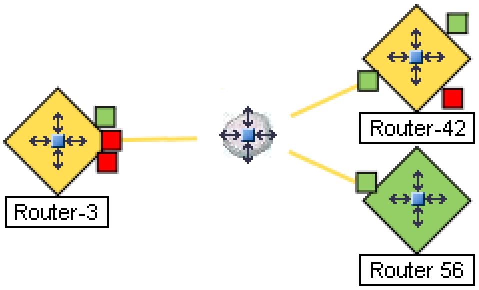
NNMi accomplishes this by following special rules for subnets with prefix lengths between 28 and 31. These special rules are called Subnet Connection Rules.

If you configure a Subnet Connection Rule, the rule independently applies to each Tenant. The members of Subnets must be unique Tenant/Node pairs (each Node assigned to only one Tenant). A Subnet Connection Rule can establish a link between the Default Tenant and another Tenant. However, links between two Tenants are not permitted *unless one of them is the Default Tenant*. See "Configure Tenants" on page 194.

These Subnet Connections Rules enable NNMi to draw arbitrary connections on maps where none would otherwise be detected. If the connection is between two nodes, NNMi draws a standard line on maps. For example:



If the connection is between more than two nodes, NNMi displays an icon:



Tip: NNMi uses Subnet Connection Rules to prevent incorrect connection calculations to Provider Edge (PE) interfaces (see Interface Capability `com.hp.nnm.capability.iface.PE`). If your network environment includes Provider

[1] Provider Edge router. The Internet Service Provider's router that receives your data on the path to your data's final desination. The Customer Edge (CE) router in your network connects to this PE.

[2] Customer Edge router. The router in your network that sends data to an Internet Service Provider's router (the Provider Edge) on the path to the data's final desination.

> Edge devices, the following HP products can provide additional valuable information for your team:
>
> - HP Network Node Manager iSPI for MPLS Software
> - HP Route Analytics Management Software (RAMS) for MPLS WAN

If you double-click the line or the icon, the Layer 2 Connection form displays and the **Topology Source** value is `SUBNETCONNECTION`.

NNMi provides a group of predefined Subnet Connection Rules (see "Subnet Connection Rules Provided by NNMi" on page 247). You can edit an existing Subnet Connection Rule or create your own (see "Configure IPv4 Subnet Connection Rules" on page 244).

If you limit Spiral Discovery to only your Discovery Seeds, NNMi uses the Subnet Connection Rules to detect connections among those devices.

If you use Auto-Discovery rules to configure Spiral Discovery, when NNMi detects a subnet prefix between 28 and 31, NNMi uses the Subnet Connection Rules:

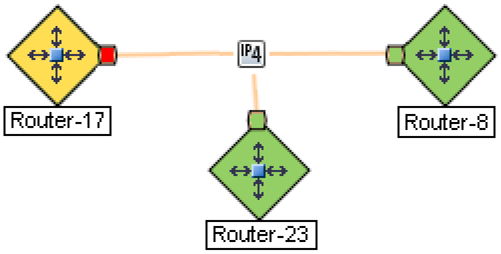
1. NNMi checks for an applicable Subnet Connection Rule (see "Subnet Connection Rules Provided by NNMi" on page 247).
2. If a match is found, Spiral Discovery checks the topology database for existing data about each IPv4 address in the subnet. If no data is found for a particular IPv4 address, NNMi issues an SNMP query to the new IPv4 address. The number of available IPv4 addresses for each valid prefix length is described in the following table:

   Valid Minimum Prefix Length Values (Subnet Mask Length)

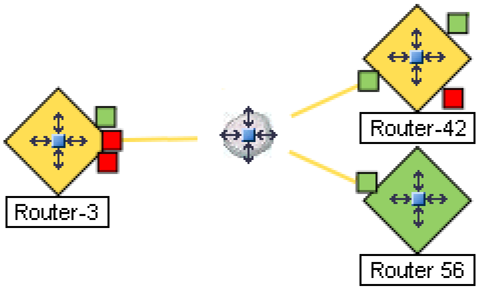
Valid Minimum IPv4 Prefix Length Values	Number of Usable IPv4 Addresses
28	14 (16-2=14)*
29	6 (8-2=6)*
30	2 (4-2=2)*
31	2

   * Two IPv4 addresses are reserved in each subnet. The first IPv4 address is used for the network itself and the last IPv4 address is reserved for broadcast.
3. NNMi checks the Excluded IP Addresses list. Any addresses in the list are dropped. For details, see "Configure an Excluded IP Addresses Filter" on page 248.
4. New IPv4 addresses that respond to SNMP are added to the topology database and available for monitoring purposes. New IPv4 addresses that do not respond to SNMP are ignored.
5. If the IPv4 address on each end of a connection has an associated interface, NNMi uses the subnet connection rule to display the connection on map views.

   In a Layer 3 Neighbor View map, if NNMi discovers an interface that is connected to more than one interface, the results of your subnet connection rule look like the following:



In a Layer 2 Neighbor View map, if NNMi discovers an interface that is connected to more than one interface, the results of your subnet connection rule look like the following:



See "Configure IPv4 Subnet Connection Rules" on page 244 to learn how to configure Subnet Connection Rules.

For details about how Spiral Discovery works:

Keep Requests to a Minimum

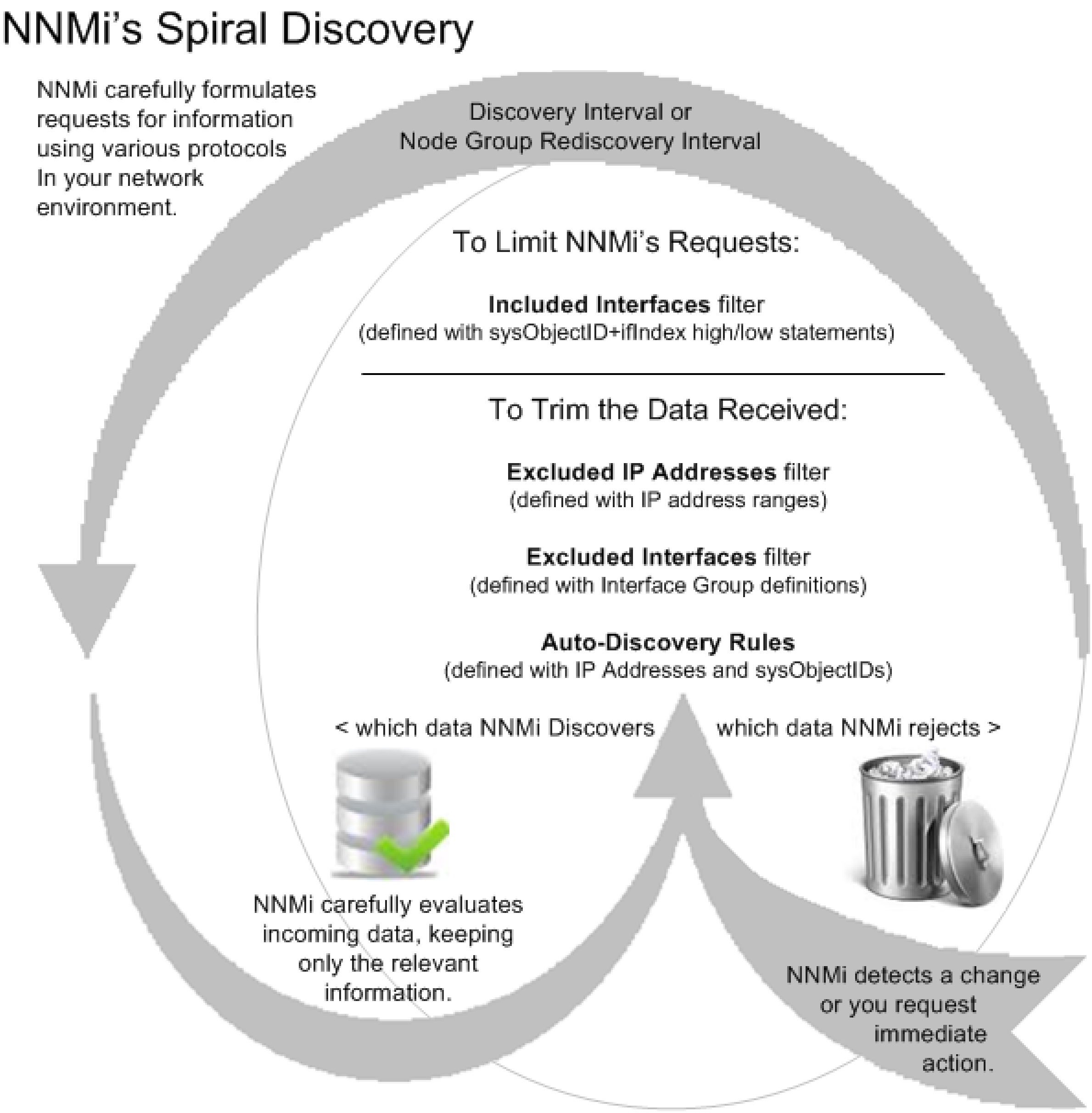
Often your network environment has devices with thousands of interfaces and you want NNMi to discover and monitor only a subset of the interfaces in these devices. To keep SNMP traffic to a minimum, use the Included Interfaces filter. This filter instructs Spiral Discovery to request information about only the subset of interfaces you specify for each vendor/make/model. See "Configure an Included Interface Ranges Filter" on page 251.

> **Tip:** You can configure NNMi to never send SNMP or ICMP requests to specific IP addresses or hostnames. See "Configuring Communication Protocol" on page 119.

To trim data from responses to Spiral Discovery's requests, use the following:

- "Configure an Excluded IP Addresses Filter" on page 248
- "Configure an Excluded Interfaces Filter" on page 254

If you choose to use Auto-Discovery within your Default Tenant's address range, Auto-Discovery Rules provide a wide range of controls. See "Example Uses of Auto-Discovery" on page 228.


NNMi's Spiral Discovery
NNMi carefully formulates requests for information using various protocols In your network environment.
Discovery Interval or Node Group Rediscovery Interval
To Limit NNMi's Requests:
Included Interfaces filter
(defined with sysObjectID+ifIndex high/low statements)
To Trim the Data Received:
Excluded IP Addresses filter
(defined with IP address ranges)
Excluded Interfaces filter
(defined with Interface Group definitions)
Auto-Discovery Rules
(defined with IP Addresses and sysObjectIDs)
< which data NNMi Discovers
which data NNMi rejects >
NNMi carefully evaluates incoming data, keeping only the relevant information.
NNMi detects a change or you request immediate action.

For details about how Spiral Discovery works:

Correct Any Misinformation

To verify Spiral Discovery's results and correct any problems, see the following:

- "Examine Discovery Results" on page 268
- "Keep Your Topology Accurate" on page 277

For details about how Spiral Discovery works:

When Does Discovery Happen?

Initial Discovery

When you add a discovery seed, NNMi immediately tries to discover that device. If discovery is not successful, NNMi tries again 10 minutes later, and continues trying. The time between each attempt is doubled until the time reaches 1 week or equals your current schedule for Rediscovery Interval. See "Configure Schedule Settings" on page 209.

> **Note:** Nodes configured as discovery seeds are always discovered and added to the topology database. If you change your mind and delete a discovery seed configuration, the node is *not* automatically deleted from the topology database. See "Delete Nodes" on page 1602.

Auto-Discovery

If you choose to use Auto-Discovery, NNMi automatically gathers Hints from each discovered Node and uses that information to find any neighboring devices within your Default Tenant's address range. This happens automatically each time a Hint is detected or an Auto-Discovery Rule includes Ping Sweep to let Auto-Discovery find every device that responds.

Rediscovery

After NNMi completes initial discovery of your network, Spiral Discovery checks for changes according to the current Schedule Settings for Rediscovery Interval:

- If a discovered Node's configuration settings or status changes, NNMi dynamically updates the database and maps to reflect the changes.

  The only exception is when non-SNMP nodes that had the same DNS hostname are changed to have separate DNS hostnames, NNMi must completely rediscover the non-SNMP nodes to correctly update the database objects (node, interface, address, connection, and incidents). The NNMi administrator must delete the old non-SNMP node object and force NNMi to rediscover the new node configurations. See "Delete Nodes" on page 1602.
- If a new node is added to your network within the Default Tenant address range and your team uses Auto-Discovery, NNMi dynamically discovers that Node, updates the topology database, and updates the maps. The details of the new node appear in the Node form. The maps reflect the new node's connectivity information.

If a node has not been rediscovered within the current Rediscovery Interval, then NNMi initiates a rediscovery after the Rediscovery Interval time frame has been reached. For example, if you set the Rediscovery Interval to 1 day, NNMi rediscovers all nodes that have not been rediscovered for other reasons after the 1 day interval has passed. NNMi strategically batches groups of nodes over time to reduce the volume of network traffic generated.

On-Going Discovery in Response to Changes

NNMi collects and analyzes data about each Node’s Tenant assignment, IP Addresses, MAC Addresses, DNS and system information to determine any change. NNMi collects this data according to the currently configured Discovery Interval value or when polling results or traps indicate that something changed.

Spiral Discovery rediscovers Nodes for a variety of reasons between the scheduled discovery interval:

- If NNMi's State Poller detects the following, NNMi rediscovers the node:
  - An SNMP-enabled Node rebooted (based on detected SNMPv2 MIB `sysUpTime` values).
  - A Node's component such as an IP address, interface, or CPU no longer exists within a previously monitored SNMP-enabled Node.
  - NNMi is configured to use SNMP for detecting `ifNumber` and `entLastChangeTime` value changes (indicating interface renumbering, new interfaces, or interfaces being removed). See instructions in the following topics for configuration instructions:
    - "Detect Interface Changes" on page 280.
    - "Default Settings for Monitoring" on page 345
    - "Node Group Settings for Monitoring" on page 391
- If certain traps are received from network devices, these traps indicate that the network topology under NNMi's management potentiality changed. Spiral Discovery rediscovers the Node involved. For example:

SNMPColdStart	CiscoColdStart	CiscoLinkDown
SNMPWarmStart	CiscoWarmStart	CiscoLinkUp
SNMPLinkDown	CiscoFRUInserted	and other vendor-equivalent traps
SNMPLinkUp	CiscoFRURemoved	

Your Rediscovery Requests

At any time, you can initiate a request to rediscover information about a previously discovered node. Select a node in any table or map view, then click the **Actions** → **Polling** → **Configuration Poll** command.

You can also use the nnmnoderediscover.ovpl or nnmconfigpoll.ovpl command to issue requests about rediscovering multiple nodes.

For details about how Spiral Discovery works:

How Is Discovery Configured?

A number of NNMi configuration settings let NNMi administrators control how Spiral Discovery works. The steps required depend on what your team wants to accomplish and the details of your network environment. See the following topic for more information:

"Determine Your Approach to Discovery" below

For details about how Spiral Discovery works:

Determine Your Approach to Discovery

Discover and monitor only the network devices that you and your team consider to be important. Take any approach that makes sense to you.

Prepare for Spiral Discovery:

- "Prerequisites for Discovery" below
- "Establish Global Defaults for Spiral Discovery" on page 201
- "Configure Schedule Settings" on page 209
- Does your network include "Overlapping Addresses in NAT Environments" on page 89?

Maintain absolute control over what is discovered.

- "Spiral Discovery of Only Seeds (all Tenants)" on page 243
- "Configure Tenants" on page 194

  Tenants are required if your network domain includes the following:

  - "Overlapping Addresses in NAT Environments" on page 89
  - "Connecting Multiple NNMi Management Servers (NNMi Advanced)" on page 93
    See also "Tenant Best Practices for Global Network Management" on page 96

Fine tune Spiral Discovery behavior:

- "Configure an Excluded IP Addresses Filter" on page 248
- "Configure an Included Interface Ranges Filter" on page 251
- "Configure an Excluded Interfaces Filter" on page 254
- "Configure IPv4 Subnet Connection Rules" on page 244 (add connections that NNMi cannot detect)

Default Tenant only: Configure Auto-Discovery to make decisions about what is discovered within the Default Tenant.

Optional. Create one or more Auto-Discovery Rules that define what is important to you and your team:

- "Configure Auto-Discovery Rules" on page 215
- "Example Uses of Auto-Discovery" on page 228

For details about how Spiral Discovery works:

Prerequisites for Discovery

For details about the required prerequisites, see the following:

Tenant definitions are required if your network domain includes the following:

- "Overlapping Addresses in NAT Environments" on page 89
- "Connecting Multiple NNMi Management Servers (NNMi Advanced)" on page 93
  See also "Tenant Best Practices for Global Network Management" on page 96

See "Configure Tenants" on page 194

NNMi uses SNMP and DNS while discovering and monitoring devices. NNMi Advanced can discover and monitor IPv6 addresses in addition to IPv4 addresses. To ensure accurate network topology information about your network environment, verify that your environment complies with the prerequisites.

SNMP Prerequisites

Spiral Discovery uses SNMP while detecting devices and connections among the devices in your network environment. NNMi also uses SNMP as part of monitoring and reporting on the health of devices in your network environment.

NNMi supports the following SNMP versions:

- SNMPv1
- SNMPv2c
- SNMPv3

NNMi uses information gathered from Routers to establish membership for Subnet connections. Make sure that important Routers in your network environment are SNMP enabled.

NNMi uses either of the following criteria to identify a Router:

- The Router responds to an SNMP query with appropriate values for `sysServices` (1.3.6.1.2.1.1.7) and `ipForwarding` (1.3.6.1.2.1.4.1). See RFC 1213, MIB-II for details.
- The Router responds to an SNMP query with an appropriate MIB-II `sysObjectID` value according to the current settings in NNMi's Device Profile configuration.

You must provide the appropriate SNMP Community Strings to NNMi. See "Configuring Communication Protocol" on page 119.

Before configuring NNMi discovery, complete the following steps:

1. Enable SNMP communication on important devices in your network (each device that you want NNMi to actively monitor).

   See the manufacturer's documentation for information about how to configure SNMP on each of your devices.

   - Establish *read community strings* for any SNMPv1 or SNMPv2c agents.
   - Establish the appropriate *User-based Security Module (USM) level of security for authentication and privacy* for any SNMPv3 agents.

2. Configure NNMi to use the appropriate *read community strings* (in the order you specify) or *USM settings* for your network environment. See "Configuring Communication Protocol" on page 119.

Well-Configured DNS Prerequisite

NNMi uses Domain Name System (DNS) to determine relationships between hostnames and IP addresses. This can result in a large number of `nslookup` requests.

> **Tip:** To improve the response time for `nslookup`, deploy a secondary DNS service on the NNMi management server or another system on the same subnet as the NNMi management server. Configure this secondary DNS service to mirror the information from the primary DNS service. Another option is to use `*/etc/hosts` instead of DNS in small environments.
>
> NNMi allows hostname as a configuration criteria for multiple features. For best results ensure that your network domain has no duplicate Domain Name System (DNS) names.

Use nslookup to Verify DNS Server Configurations

Verify that your DNS servers are well configured to prevent long delays when resolving `nslookup` requests. This means the DNS server responding to NNMi `nslookup` requests has these qualities:

- The DNS server is an authoritative server and does not forward DNS requests.
- The DNS server has consistent *hostname-to-IP address* mappings and *IP address-to-hostname* mappings.
- If your network uses multiple DNS servers, all respond consistently to any particular `nslookup` request.

> **Caution:** Round-robin DNS (used to do load balancing of web application servers) is not appropriate because any given hostname can map to different IP addresses over time.

On the NNMi management server, verify that the following configuration settings in your environment:

- **All operating systems**: Locate your `*/etc/hosts` file and ensure that the host file contains a minimum of two entries. When an `nslookup` command is not successful, this file takes over:

  `127.0.0.1` (loopback loghost) or `::1`
  `<NNMi_server_address>` (the IP address of the NNMi management server)

  If your NNMi management server participates in a high availability (HA) environment, the virtual server name and IP-address is required in the `*/etc/hosts` file in addition to the physical server name and IP-address.

  Windows: The following registry key determines the location of this file:

  ```
  \HKEY_LOCAL_
     MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\DataBa
     sePath
  ```

  UNIX: This file is in the `/etc` directory.

- **Windows:** Use the Control Panel to navigate to your Network and Internet Connections configuration, Network Connections, Local Area Connections, Support tab, and click the Details button. Verify that all identified DNS servers provide consistent *hostname-to-IP address* mappings and *IP address-to-hostname* mappings.
- **UNIX:** Ensure that the `nslookup` search path resolves to the `nsswitch.conf` file. See the *nsswitch.conf(4)* manpage that was provided with your operating system. Verify that all

identified DNS servers provide consistent *hostname-to-IP address* mappings and *IP address-to-hostname* mappings.

Exclude Problem Devices from nslookup

You can populate two files that instruct `nslookup` to exclude certain addresses. The benefits of doing this are as follows:

- Speed up Spiral Discovery.
- Keep network traffic generated by NNMi to a minimum.

If you know there are problems with the DNS configuration in your network domain (hostnames or addresses that do not resolve properly), instruct NNMi to avoid `nslookup` requests for unimportant devices.

To identify problem devices, create the following two files before configuring NNMi discovery. NNMi never issues a DNS request for hostnames or IP addresses identified in these files:

- hostnolookup.conf — Enter fully-qualified hostnames or wildcards that identify groups of hostnames.
- ipnolookup.conf — Enter fully-qualified IP addresses or wildcards that identify groups of IP addresses.

Use an ASCII editor to populate the files. Place the files in the following location on the NNMi management server:

- **Windows**:
  `%NnmDataDir%\shared\nnm\conf\`
- **UNIX**:
  `/var/opt/OV/shared/nnm/conf/`

IPv6 Addresses Prerequisite (*NNMi Advanced*)

To discover and monitor both IPv4 and IPv6 IP addresses, the settings in the `nms-jboss.properties` file must first be configured.

NNMi Advanced. To enable NNMi to access and monitor IPv6 addresses, see the "Configuring NNMi Advanced for IPv6" chapter in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`. One of the configuration steps explains how to make changes to the `nms-jboss.properties` file settings.

To check NNMi for the status of the IPv6 feature in your networking environment, click **Help** → **System Information** and navigate to the **Server** tab.

In the Management Server section, locate the following attributes and their current values:

- **IPv6 Address:**`<`*`IP address value`*`>` (Indicates whether the NNMi management server has an IPv6 address.)
- **IPv6 Management**:

  - Enabled = NNMi Advanced currently discovers and monitors IPv6 addresses.
  - Disabled = See the *HP Network Node Manager i Software Deployment Reference* to configure NNMi for IPv6 addresses.
  - Not Licensed = Requires an upgrade to the *NNMi Advanced* license.

- **IPv6 Communication**:
  - Enabled = NNMi Advanced currently discovers and monitors IPv6 addresses.
  - Disabled = See the *HP Network Node Manager i Software Deployment Reference* to configure NNMi for IPv6 addresses.
  - Not available = An IPv6 address does not exist on the NNMi management server.

The NNMi administrator configures how NNMi requests each SNMP agents' Management Address (see "Configure Default SNMP, Management Address, and ICMP Settings" on page 120):

1. **Configuration** workspace → **Communication Configuration** → in the **Management Address Selection** area
2. > **Tip:** This attribute does not appear under Management Address Selection Settings until the NNMi Administrator follows the instructions for enabling IPv6 in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.

   IP Version Preference: Select one of the following to influence Spiral Discovery's evaluation of *newly discovered nodes*. Previously established Management Addresses will not change if you modify this IP Version Preference setting:

   - IPv4
   - IPv6
   - Any (either IPv4 or IPv6)

   > **Tip:** When set to Any, Spiral Discovery gives preference to IPv4 addresses when determining the Management Address of *newly discovered nodes*.

Overlapping Address Mapping

Overlapping Address Mapping can help you manage areas in your network that are using address translation protocols, resulting in overlapping and duplicate addresses. See "Overlapping Addresses in NAT Environments" on page 89 for more information about possible network configurations.

> **Caution:** If you are configuring NNMi for areas of your network management domain that use *dynamic* Network Address Translation (NAT) or *dynamic* Port Address Translation (PAT/NAPT), the information in this section does not apply.

If *static* Network Address Translation (NAT) is part of your network management domain, and the NNMi management server is outside of that static NAT domain, you can use Overlapping Address

Mapping to configure NNMi for displaying the NAT *external IP address* (public address). This value appears in the Mapped Address attribute of the IP Address form for the identified Tenant / NAT *internal IP address* pair. This configuration setting is also important for node monitoring,

Your network domain's *static* NAT configuration might apply to public IP addresses, private IP addresses, or both.

Network administrators use address translation protocols as a strategy in the following situations:

- When preventing direct Internet access to increase security.
- When not enough public IPv4 addresses are available within their network domain. Packets from the private IP address range are not permitted on the public Internet unless they pass through a protocol that converts the private IP address to a valid public address.

To configure NNMi to display the *static* NAT *external IP address* in the Mapped Address attribute of the IP Address form for the identified Tenant / NAT *internal IP address* pair, you must configure each domain as a unique Tenant. See "Configure Tenants" on page 194.

Then do one of the following:

- Use the "Overlapping Address Mapping Form" below.
- Use the command line tool nnmloadipmappings.ovpl.

Tip: To see the results of all mappings, use the Inventory: IP Addresses (All Attributes) view.

Private IP Address Ranges

The Internet Engineering Task Force (IETF) and Internet Assigned Numbers Authority (IANA)'s reserved the following IP address ranges for private networks, for example enterprise local area networks (LANs), corporate offices, or residential networks.

IPv4 private address ranges (RFC 1918):

- 10.0.0.0 – 10.255.255.255 (24-bit block)
- 172.16.0.0 – 172.31.255.255 (20-bit block)
- 192.168.0.0 – 192.168.255.255 (16-bit block)

IPv6 private address ranges:

- fc00::/7 address block = RFC 4193 Unique Local Addresses (ULA)
- fec0::/10 address block = deprecated (RFC 3879)

Overlapping Address Mapping Form

If *static* Network Address Translation (NAT) is part of your network management domain, and the NNMi management server is outside of that static NAT domain, you can use Overlapping Address Mapping to configure NNMi for the following:

- Populate the Mapped Address attribute of the IP Address form for the identified Tenant / NAT *internal IP address*. This Mapped Address attribute displays the corresponding NAT *external IP address* (public address).
- Ensure that in the following special cases, Spiral Discovery successfully detects changes:

- The Communication Configuration you establish for a Node enables the **Enable SNMP Address Rediscovery** ☑ attribute ("Configuring Communication Protocol" on page 119). This setting instructs NNMi to search for a new SNMP agent for the Node if the currently configured SNMP agent stops communicating for any reason (rather than waiting for the SNMP agent to come back online).
- The Monitoring Configuration you establish for a Node enables the **Enable IP Address Fault Polling** ☑ attribute ("Monitoring Network Health" on page 340). This setting instructs NNMi to use ICMP. When using ICMP for this purpose, the Overlapping IP Address Mapping is required for each monitored internal address within the Static NAT.

Your network domains might use *static* NAT for duplicate addresses in enterprise local area networks (LANs), corporate offices, or residential networks. See "Overlapping Addresses in NAT Environments" on page 89 for more information about possible network configurations.

> **Note:** If you are configuring NNMi for areas of your network management domain that use *dynamic* Network Address Translation (NAT) or *dynamic* Port Address Translation (PAT/NAPT), do not use this form. For more information:

To configure NNMi to display *static* Network Address Translation (NAT) *external IP address* in the IP Address form, do the following:

> **Tip:** There is also a command line tool for this task nnmloadipmappings.ovpl.

1. Prerequisite: Configure each network management domain as a unique Tenant. See "Configure Tenants" on the next page.
2. Navigate to the **Overlapping Address Mapping** view.
   a. From the workspace navigation panel, select the 🔧 **Configuration** workspace.
   b. Select **Discovery**.
   c. Select **Overlapping Address Mapping**.
3. Do one of the following:
   - To create a new configuration, click the ✱ **New** icon.
   - To edit an existing configuration, double-click the Overlapping IP Address Mapping definition you want to edit.
   - To delete a configuration, select the Overlapping IP Address Mapping definition you want to delete and click the ✖ Delete icon.
4. Make your configuration choices, all three settings are required. (See the Overlapping IP Address Mapping Attributes table.)
5. Click 💾 **Save and Close**.

Note: If you reassign a Node from one Tenant to another Tenant, this setting does not automatically update.

Overlapping Address Mapping Attributes

Attribute	Description
Tenant	Designate which Tenant owns the Addresses you are mapping. See "Configure Tenants" below. Click the Lookup icon and do one of the following: • To select an existing Tenant configuration, click the **Quick Find** icon • To create a new configuration, click the **New** icon. **Note:** This attribute value does not automatically update if the NNMi administrator reassigns the Node to another Tenant.
External Address	Provide the appropriate substitute address configured in *static* Network Address Translation (NAT) for the Internal Address (next value). The address you provide here shows up in the IP Address form's **Mapped Address** attribute if NNMi discovers the designated Tenant / Internal Address pair.
Internal Address	Provide the address that requires mapping. The External Address you map to this address appears in the IP Address form's **Mapped Address** attribute.

Configure Tenants

For details about configuring Tenants, see the following:

NNMi administrators use Tenant settings to accomplish the following:

- Identify overlapping address domains in your network so NNMi can avoid duplicate address problems. An unique Tenant is required for each group of devices configured to use any of the following *address translation protocols:*
    - *Static* Network Address Translation (NAT)
    - *Dynamic* Network Address Translation (NAT)
    - *Dynamic* Port Address Translation (PAT/NAPT)

  For more information:
- Determine precise groups of Nodes when your Subnet mask strategy fails. NNMi uses the Tenant:Subnet pair to identify each group of Nodes. You can manage groups of Nodes even when deployed Subnets conflict within your network management domain. Nodes within a Subnet can belong to different Tenants. NNMi calculates each Tenant's Subnets independently.

NNMi administrators can easily change an Node's Tenant assignment, see"Change Tenant Assignment for a Node" on page 289.

If you configure a Subnet Connection Rule, the rule independently applies to each Tenant. The members of Subnets must be unique Tenant/Node pairs (each Node assigned to only one Tenant). A Subnet Connection Rule can establish a link between the Default Tenant and another Tenant. However, links between two Tenants are not allowed *unless one of them is the Default Tenant*. See "Consider IPv4 Subnet Connection Rules" on page 181.

- Control the connections NNMi identifies among Nodes.

  Devices that belong to the Default Tenant can have Layer 2 Connections to any device in any Tenant. Devices within any Tenant *other than* Default Tenant can have Layer 2 Connections *only* to devices within the same Tenant or the Default Tenant.

- Establish the relationship between Provider Edge (**PE**[1]) devices and Customer Edge (**CE**[2]) devices. Assign Provider Edge (**PE**[3]) devices to the Default Tenant. Assign Customer Edge (**CE**[4]) devices to a Tenant created by the NNMi administrator.

- Assign any infrastructure device that interconnects multiple Network Address Translation (**NAT**[5]) domains (such as a NAT gateway) to the Default Tenant. This ensures that NNMi displays the Layer 2 Connections your team and customers need to see.

- Identify members of a Router Redundancy Group (all members must be assigned to the same Tenant, multiple Router Redundancy Groups can belong to the same Tenant).

- *Global Network Management:* Manage the Tenant and Security Group settings for Nodes replicated from Regional Managers to the Global Manager. See "About Multi-Tenancy and Global Network Management" on page 95 and "Tenant Best Practices for Global Network Management" on page 96.

  Tenant definitions can be exported/imported among all NNMi management servers. See "Export/Import Behavior and Dependencies" on page 1579.

- Conveniently assign an *Initial Discovery Security Group* to Seeds before discovery.

  NNMi administrators can change a node's Tenant or Security Group assignment at any time. See "Specify Discovery Seeds" on page 256 for more information.

> **Note:** *Auto-Discovery* is available only for the Default Tenant. Each automatically discovered node is assigned to the Default Tenant (and the *Initial Discovery Security Group* currently configured for newly discovered nodes in the Default Tenant).

[1] Provider Edge router. The Internet Service Provider's router that receives your data on the path to your data's final desination. The Customer Edge (CE) router in your network connects to this PE.

[2] Customer Edge router. The router in your network that sends data to an Internet Service Provider's router (the Provider Edge) on the path to the data's final desination.

[3] Provider Edge router. The Internet Service Provider's router that receives your data on the path to your data's final desination. The Customer Edge (CE) router in your network connects to this PE.

[4] Customer Edge router. The router in your network that sends data to an Internet Service Provider's router (the Provider Edge) on the path to the data's final desination.

[5] Network Address Translation. NNMi supports the following protocols: Static Network Address Translation, Dynamic Network Address Translation, Dynamic Port Address Translation.

> Devices within the Default Security Group are visible from all views. To control access to a device, assign that device to a Security Group other than Default Security Group.

- Identify logical groups of Nodes for any purpose, for example to identify the resources assigned to a specific customer or to identify specific areas of your network or to identify company sites.
- Create Node Groups based on Tenant attribute values. See "Specify Node Group Additional Filters" on page 298 for more information about Node Group filters.
- Configure Incidents based on Tenant attribute values. See "Custom Incident Attributes Provided by NNMi (Information for Administrators)" on page 647.

Use the Tenant Form

NNMi's Tenant configuration settings are useful for a variety of situations. Review the Tenant information so you know about all your options. See "Configure Tenants" on page 194 for more information.

NNMi provides a Tenant named *Default Tenant*. NNMi administrators can create additional Tenant objects as needed. A discovered node that is not specifically assigned to a particular Tenant, automatically becomes a member of the Default Tenant. NNMi administrators can change a Node's Tenant assignment at any time. Depending on the network environment, the NNMi administrator decides whether or not additional Tenants are needed.

When additional Tenants are defined, Tenant assignments are visible in the Node form's Basic Attributes and in the Tenants column of the Inventory > Nodes view.

Devices that belong to the Default Tenant can have Layer 2 Connections to any device in any Tenant. Devices within any Tenant *other than* Default Tenant can have Layer 2 Connections *only* to devices within the same Tenant or the Default Tenant.

> **Tip:** Assign any infrastructure device that interconnects multiple NAT domains (such as a NAT gateway) to the Default Tenant. This ensures that NNMi displays the Layer 2 Connections your team and customers need to see.

NNMi administrators can easily change a Node's Tenant assignment at any time, see "Change Tenant Assignment for a Node" on page 289.

To configure a Tenant, do the following:

1. Navigate to the **Tenants** view.
   a. From the workspace navigation panel, select the 🔧 **Configuration** workspace.
   b. Select **Discovery**.
   c. Select **Tenants**.
   d. Do one of the following:
      - To create a new configuration, click the ✱ **New** icon.
      - To edit an existing configuration, double-click the Tenant definition you want to edit.

- To delete a configuration, select the Tenant definition you want to delete and click the ✖ Delete icon.

2. Make your configuration choices. (See the Tenant Attributes table.)
3. Click 💾 **Save and Close**.
4. Best practice: If the Tenant participates in a Global Network Management environment, replicate the Tenant configuration to the Global Manager.
5. The Tenant attribute displays on each Node form (use the drop-down list to change the assigned Tenant attribute value, or use nnmsecurity.ovpl).

   NNMi administrators use the Tenant object to do the following:

   - Associate a Tenant with each Discovery seed - before discovery ("Specify Discovery Seeds" on page 256).
   - Enable monitoring of nodes with addresses provided by *static* Network Address Translation (NAT), *dynamic* Network Address Translation (NAT), or *dynamic*Port Address Translation (PAT/NAPT), see "Overlapping Addresses in NAT Environments" on page 89.
   - "Specify Node Group Additional Filters" on page 298
   - Populate the Tenant attribute on the Node form (see "Custom Incident Attributes Provided by NNMi (Information for Administrators)" on page 647).

Tenant Attributes

Attribute	Description
Name	Enter the name that uniquely identifies this Tenant. If your team uses NNMi's Global Network Management feature, before choosing a name, see "About Multi-Tenancy and Global Network Management" on page 95. **Note:** You must enter a Name value.
UUID	NNMi assigns a Universally Unique Object Identifier to the Tenant. This UUID is unique across all databases.
Description	Type a maximum of 2048 characters to describe this User Group. Alpha-numeric, spaces, and special characters (~ ! @ # $ % ^ & * ( ) _+ -) are permitted.
Initial Discovery Security Group	The Initial Discovery Security Group specifies the Security Group assigned to any *seed* associated with this Tenant object (before discovery). See "Tenant and Initial Discovery Security Group Assignments" on the next page and "About Security Groups" on page 515. **Caution:** Devices within the *Default Security Group* are visible from all views. To control access to a device, assign that device to a Security Group other than Default Security Group. NNMi administrators can assign each Node within one Tenant to a different Security Group. In the **Initial Discovery Security Group** attribute, do one of the following:

Tenant Attributes, continued

Attribute	Description
	• To change the Initial Discovery Security Group, begin to type a valid Security Group Name and use the auto-complete feature to select the Security Group. **Tip:** You can also select **Quick Find** from the Lookup field drop-down list. This option is useful when you want to see more than the Security Group Name when determining which Security Group to select. • To create a new Initial Discovery Security Group, in the Lookup field, select the ✱ **New** icon.

Related Topics

"Troubleshoot NNMi Access" on page 582

"About Security Groups" on page 515

Tenant and Initial Discovery Security Group Assignments

When NNMi discovers nodes in your network environment, Tenant and Security Group settings are established in the following manner:

- **Discovery Seeds**: If Nodes are discovered as Discovery seeds, the NNMi administrator specifies a Tenant for each Discovery Seed. See "Specify Discovery Seeds" on page 256. When NNMi administrators define a Tenant, they specify an **Initial Discovery Security Group**. Any newly discovered Node within the defined Tenant is assigned to this Security Group. NNMi administrators can change either the node's Tenant or Security Group assignment or both at any time.

  Nodes assigned to the *Default Security Group* are visible from all views. To control access to a device, assign that device to a Security Group other than Default Security Group.

  Nodes within one Tenant can each be assigned to different Security Groups, and Nodes within one Security Group each be assigned to different Tenants.

- **Auto-Discovery for Default Tenant**: When you configure Auto-Discovery Rules, NNMi assigns any Nodes discovered using those Auto-Discovery Rules to the *Default Tenant* and whichever Security Group is currently configured as the Default Tenant's Initial Discovery Security Group setting (the *Default Security Group* out-of-box). See "Configure Tenants" on page 194 .

Global Network Management: Regional Managers forward information about Nodes to the Global Manager. The Global Manager's copy of the Node object has the same Tenant assignment as the Regional Manager's record of that Node.

In a Global Network Management environment, best practice is to have the NNMi administrators for the Global Manager and all Regional Managers agree to a predefined list of Tenant names. Those Tenants would be defined on the Regional Managers, the Tenant definitions exported, and those Tenant definitions imported onto the Global Manager (thus ensuring that the UUID and name value for each Tenant match on both NNMi management servers). The NNMi administrator on the Global

Manager update their Tenant definitions to assign Initial Discovery Security Group values that make sense for the Global Manager's team. See "About Multi-Tenancy and Global Network Management" on page 95 for more information.

Note: If a Regional Manager forwards information about a Node to the Global Manager, and that Node is assigned to a Tenant object that does not exist on the Global Manager, NNMi creates a Tenant with the UUID and name from the Regional Manager, but creates a new Security Group with that Tenant name (does not duplicate the Regional Manager's setting for that Tenant's *Initial Discovery Security Group* setting). NNMi maps that new Security Group to the following:

- User Group = NNMi Administrator
- Object Access Privilege = Object Administrator

The Global Manager's NNMi administrator can assign a *different* Initial Discovery Security Group to a Tenant definition at any time. From that point onward, the NNMi Global Manager uses that new Initial Discovery Security Group setting when creating new nodes within that Tenant.

Consider setting up your Security Configuration so that all newly-discovered Nodes belong to a Security Group that is mapped to User Group = NNMi Administrators . Those Nodes will be visible only to NNMi administrators until an NNMi administrator intentionally moves the node into a Security Group that is also visible to the appropriate NNMi operator or guest.

Tenant assignments determine L2 Connections between nodes on NNMi maps, and are useful for identifying groups of nodes within your network environment (for example, subnets, router redundancy groups, and Node Groups). Security Group assignments enable NNMi administrators to restrict the visibility of nodes within the NNMi console to specific User Groups. See "Configuring Security" on page 503 for more information.

Configure Discovery

NNMi uses Simple Network Management Protocol (SNMP read-only queries), and a variety of communication protocols to discover the devices within the network management domain that you define. See "How Spiral Discovery Works" on page 176 for more information.

NNMi provides one predefined Tenant, the *Default Tenant*. Each Node must be assigned to a Tenant. If you choose to use Auto-Discovery Rules, those rules apply only to the nodes within the Default Tenant. All other Discovery configuration settings apply to the nodes within all Tenants.

Tip: *Optional*. Establish additional Tenant configurations to identify overlapping address domains or to fine tune Layer 2 connections between devices in your network domain. For details, see "Configure Tenants" on page 194.

Devices that belong to the Default Tenant can have Layer 2 Connections to any device in any Tenant. Devices within any Tenant *other than* Default Tenant can have Layer 2 Connections *only* to devices within the same Tenant or the Default Tenant.

Discovery Configuration Tasks

Task	How
"Prerequisites for Discovery" on page 187	Complete all prerequisites..
"Establish Global Defaults for Spiral Discovery" on the next page	Use the Global Control panel to review the default values that NNMi provides. Determine if those defaults work for Spiral Discovery in your network environment. NNMi administrators can change the default settings at any time.
"Configure Schedule Settings" on page 209	Use the **Schedule Settings** tab to review the default values that NNMi provides for Spiral Discovery's Schedule Settings. Determine if those defaults work for Spiral Discovery in your network environment. NNMi administrators can change the default settings at any time.
"Configure Auto-Discovery Rules" on page 215	*Optional*. Use the **Auto-Discovery Rules** tab to to specify ranges of IP addresses or MIB-II `sysObjectID` values (or both) that you want NNMi to automatically discover or never discover within the Default Tenant. **Note:** NNMi assigns each node found by Auto-Discovery to the *Default Tenant* (and whichever Security Group attribute value is currently configured for the Default Tenant = the *Default Security Group* out-of-box). See "Configure Tenants" on page 194 and "About Security Groups" on page 515 for more information.
"Configure IPv4 Subnet Connection Rules" on page 244	*Optional*. Use the IPv4 **Subnet Connection Rules** tab to establish connections between interfaces on devices that *do not respond* to Layer 2 *discovery protocols* (see the list of Topology Source protocols in Layer 2 Connection Form). For example, use Subnet Connection Rules to establish connections to WAN edge devices that NNMi would not automatically detect.
"Configure an Excluded IP Addresses Filter" on page 248	*Optional*. Use the **Excluded IP Addresses** tab to provide a list of specific addresses or ranges of addresses that you want NNMi to *never* discover or monitor. This filter applies to all nodes in all Tenants.
"Configure an Included Interface Ranges Filter" on page 251	*Optional*. Use the **Included Interface Ranges** tab to provide a MIB-II `sysObjectID` list and designate which Interfaces within devices of that type NNMi is permitted to discovery (all other interfaces within devices meeting the MIB-II `sysObjectID` criteria are ignored). This filter applies to all nodes in all Tenants.
"Configure an Excluded	*Optional*. Use the **Excluded Interfaces** tab to provide a list of specific interfaces that you want NNMi to *never* discover or monitor.

Discovery Configuration Tasks , continued

Task	How
Interfaces Filter" on page 254	This filter applies to all nodes in all Tenants.
"Choose Techniques to Launch Discovery" on page 240	You control Spiral Discovery's starting points: • Any Tenant: Use the **Seeds** Configuration workspace to specify the nodes to be discovered. **Tip:** Use the Seeds workspace to verify that NNMi successfully located each Discovery Seed that you provided. See "Discovery Seed Results" on page 270. • Default Tenant only: If you choose to use Auto-Discovery, there are two choices for launching discovery. Seeds can provide the starting points from which Auto-Discovery gathers information about neighboring devices to expand discovery. Or Ping Sweep settings can enable Auto-Discovery to find any device that responds to Ping commands.

Establish Global Defaults for Spiral Discovery

Decide if you want to change any of the global default settings for Spiral Discovery:

The Global Defaults determine the following:

- Enable/Disable ATM / Frame Relay Interfaces for Performance Monitoring.
- Enable/Disable Ping Sweep for Auto-Discovery of IPv4 addresses.
- Configure the strategy NNMi uses to determine Node Names.
- Specify zero or one Node Group from which Spiral Discovery will *ignore* the Forwarding Database (FDB) data when calculating Layer 2 Connections (the FDB data is still included in other calculations).

Configure Discovery of ATM/Frame Relay Interfaces

(*NNM iSPI Performance for Metrics* only) If your network environment includes devices that are using Asynchronous Transfer Mode (ATM) or Frame Relay protocols, NNM iSPI Performance for Metrics can provide useful information about network activity that is using those protocols.

To enable/disable Discovery of ATM/Frame Relay Interfaces:

1. Navigate to the **Discovery Configuration** form.
   1. From the workspace navigation panel, select the **Configuration** workspace.
   2. Expand **Discovery**.
   3. Select **Discovery Configuration**.
2. Locate the **Global Control** settings.
3. Specify the ATM/Frame Relay Discovery setting.

Global Control Attributes

Name	Description
Enable Discovery of ATM/Frame Relay Interfaces for Performance Monitoring	If your team installed *HP Network Node Manager iSPI Performance for Metrics Software*: If ☑ enabled, this attribute extends the range of data that NNMi gathers for ATM and Frame Relay interfaces. If □ disabled NNMi does not discover and gather the extended ATM and Frame Relay data that NNM iSPI Performance for Metrics uses for reporting purposes. See also "Configure NNMi Monitoring Behavior" on page 340 for information about the Monitoring Configuration settings for *Enable ATM Interface Performance Polling* and *Enable Frame Relay Interface Performance Polling* (Default, Node, or Interface settings).

4. Click **Save and Close** to apply your changes.

Configure Ping Sweep (override for all Auto-Discovery Rules)

***Default Tenant* only**: You have two choices for Auto-Discovery starting points. Use either or both to best advantage in your network environment:

- Discovery Seeds: You designate specific hostnames (*not case-sensitive*) or IP addresses where Auto-Discovery starts gathering neighbor information.

  For details see "Discovery Seeds for Auto-Discovery in Default Tenant" on page 241. For information about creating Discovery Seeds. See "Specify Discovery Seeds" on page 256.

- Ping Sweep: NNMi issues ICMP pings to certain addresses to find new nodes. For details, see "Ping Sweep for Auto-Discovery in Default Tenant" on page 242.

  IPv4 addresses only: In Wide Area Networks (WANs) such as ATM, Frame Relay, and Point-to-Point (where ARP cache is not available), the optional "Ping Sweep for Auto-Discovery in Default Tenant" on page 242 feature locates nodes for Auto-Discovery to use when gathering neighbor information and evaluating connections between nodes, see also "Consider IPv4 Subnet Connection Rules" on page 181.

> **Note:** Ping Sweep works only with IPv4 addresses. All nodes discovered using Auto-Discovery are assigned to the *Default Tenant*.

Ping Sweep uses the current default ICMP interval and timeout settings from the Communications Configuration settings. See "Configure Default SNMP, Management Address, and ICMP Settings" on page 120.

To configure the global Auto-Discovery setting for Ping Sweep:

1. Navigate to the **Discovery Configuration** form.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Expand **Discovery**.
   c. Select **Discovery Configuration**.
2. Navigate to the **Global Control** settings.
3. Designate the global setting for **Ping Sweep**. Your choice determines how Auto-Discovery uses ICMP ping commands for the discovery process in your network environment:
   - **Each Rule (as configured)**— The instructions for Ping Sweep within each Auto-Discovery Rule configuration are followed exactly.

     To configure Ping Sweep for a specific Auto-Discovery Rule, see "IP Address Ranges for the Auto-Discovery Rule" on page 221.
   - **All Rules**— Ping Sweep is applied for all of your current Auto-Discovery Rules. This overrides the Ping Sweep settings within each rule. Spiral Discovery issues the initial round of Ping Sweep commands when you click Save and Close.
   - **None of the Rules**— Ping Sweep is not used for any of your current Auto-Discovery Rules. This overrides the Ping Sweep settings within each rule. This is useful to temporarily suspend issuing any ping commands within your network.

   > **Note:** If things do not work as expected, check whether ICMP is enabled (see if "Communication Region Form" on page 137).
4. Designate the **Sweep Interval** (days/hours) that controls how often Auto-Discovery reissues ICMP Ping for each address. The minimum Sweep Interval setting is 1 hour. Maximum 99 days.
5. Click **Save and Close**. Spiral Discovery issues the initial round of Ping Sweep commands when you click Save and Close.

Configure the Node Name Strategy

For more details about how NNMi determines the Node name, see the following:

NNMi administrators control how the Name attribute on the Node form is populated. To resolve issues about choosing the Name value, NNMi follows a sequence of rules. If NNMi is unable to determine a Name based on your three choices, the node name is determined using the NNMi factory defaults for these three choices (see list in step 3).

The node Name shows up beneath the node symbol on the maps and in the Name column on table views.

To control how node names are determined for your network devices:

1. Navigate to the **Discovery Configuration** form.
   a. From the workspace navigation panel, select the 🔧 **Configuration** workspace.
   b. Expand **Discovery**.
   c. Select **Discovery Configuration**.
2. Locate the **Node Name Resolution** attributes on the left side of the form (see table).
3. Specify the three-level hierarchy for node naming decisions.

   Short name and full name are related. The short name is everything before the first period in the full name. For example, full name `cisco5500.abc.example.com` and the short name `cisco5500`.

   > **Note:** NNMi administrators can use NNMi property file settings to change the way NNMi determines Hostname values:
   >
   > - `nms-topology.properties` file settings:
   >   If DNS is the source of the Node's Hostname, there are three choices. By default NNMi uses the exact Hostname from your network configuration. It is possible to change NNMi behavior to convert Hostnames to all uppercase or all lowercase. See the "Modifying NNMi Normalization Properties" section of the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.
   > - `nms-disco.properties` file settings:
   >   The Hostname is either requested from the Node's lowest loopback interface IP address that resolves to a Hostname or requested from the Node's designated Management Address (SNMP agent address). With either choice, when no IP address resolves to a Hostname, the IP address itself becomes the Hostname. See the "Maintaining NNMi" chapter of the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.

   Select among the following choices. Use each choice only one time:

   - **Short DNS Name** – (*first by default*) Use the group of characters before the first period in your in-house DNS naming standards. See "Discovery Node Name Choices" on the next page for possible issues with using DNS names.
   - **Fully Qualified DNS Name** – Use the full in-house DNS naming standards.
   - **Short sysName** – (*second by default*) Use the group of characters before the first period in the current MIB-II `sysName` value established by the administrator for each SNMP enabled

device. See "Discovery Node Name Choices" below for possible issues with using `sysName`.

- **Full sysName** – Use the full MIB-II `sysName` value established by the administrator for each SNMP enabled device.
- **IP Address** – (*third by default*) Use the IP address. If the node responds to SNMP, the SNMP Management Address is used. For non-SNMP nodes, name is set to either a discovery seed address associated with this node or a neighbor address gathered by Auto-Discovery along the path to this node.

> **Note:** NNMi administrators can make choices that limit NNMi's use of IP addresses:
>
> - IP addresses in the Excluded IP Addresses filter are never used, Spiral Discovery skips those addresses. See Configure an Excluded IP Addresses Filter.
> - *NNMi Advanced:* IPv4 or IPv6 addresses are used according to configuration choices. See IPv6 Addresses Prerequisite.

4. Click **Save and Close**. Spiral Discovery implements your changes during the next regularly scheduled discovery interval. To apply the changes immediately, use **Actions** → **Polling** → **Configuration Poll**. See Using Actions to Perform Tasks for more information.

> **Tip:** You can right-click any object in a table or map view to access the **Actions** menu.

Node Name Resolution Settings

Attribute	Description
First Choice	Click the drop-down list and choose the predefined node name strategy you want discovery to use first.
Second Choice	Click the drop-down list and choose the predefined node name strategy you want discovery to use if the first choice fails.
Third Choice	Click the drop-down list and choose the predefined node name strategy you want discovery to use if the second choice fails.

Discovery Node Name Choices

Control how the **Name** attribute on node forms is populated during discovery. This Name value is used to identify the object in NNMi maps and table views. You specify a hierarchy for discovery to use. You configure three levels in the hierarchy. See "Node Name Decision Tree" on page 207.

You can designate any of the following for each level of the node Name decision hierarchy:

- **DNS Names**. Discovery uses the results of hostname resolution.

  NNMi follows a set of rules to dynamically generate the value stored in the NNMi database for each Node's Hostname. Click here for details.

  - If the Node supports SNMP, NNMi requests the Hostname using the IP Address of the associated SNMP agent (the Management Address attribute value on the Node form).

  When the NNMi administrator chooses **Enable SNMP Address Rediscovery** ☑ in the Communication Configuration:

  - If the SNMP Agent does not respond, NNMi checks for another Management Address to request the Hostname, and the Hostname could change.
  - If the SNMP Agent associated with the node changes, the Management Address and Hostname could change.

  When the NNMi administrator disables **Enable SNMP Address Rediscovery** □ in the Communication Configuration:

  - If the SNMP Agent does not respond, NNMi uses the previously gathered Management Address attribute value to request the Hostname.
  - If the SNMP Agent associated with the node changes, NNMi uses the previously gathered Management Address attribute value to request the Hostname.

- If the Node does not support SNMP, no Management Address is available. NNMi requests a Hostname starting with the lowest IP Address associated with the node (a Discovery Seed value or an IP address value gathered from a neighboring device). NNMi uses the first Hostname provided. The Hostname might change during a future discovery cycle.

> **Note:** NNMi administrators can use NNMi property file settings to change the way NNMi determines Hostname values:
>
> - `nms-topology.properties` file settings:
>   If DNS is the source of the Node's Hostname, there are three choices. By default NNMi uses the exact Hostname from your network configuration. It is possible to change NNMi behavior to convert Hostnames to all uppercase or all lowercase. See the "Modifying NNMi Normalization Properties" section of the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.
> - `nms-disco.properties` file settings:
>   The Hostname is either requested from the Node's lowest loopback interface IP address that resolves to a Hostname or requested from the Node's designated Management Address (SNMP agent address). With either choice, when no IP address resolves to a Hostname, the IP address itself becomes the Hostname. See the "Maintaining NNMi" chapter of the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.

- **MIB-II sysName Values**. Device administrators set the `sysName`. Discovery avoids populating the NNMi database with multiple devices having the same manufacturer's default `sysName`. If a `sysName` matches or starts with the manufacturer's default factory setting (case-sensitive), discovery ignores `sysName` as a choice for the Name attribute of the node. NNMi ships with a Device Profile for each device type (vendor/make/model). The Device Profile includes a record of the manufacturer's default `sysName`.

  Caution: You can override this choice using the Device Profile's Advanced settings, Never Use `sysName` attribute. See "Configure Device Profiles" on page 292 for more information.

- **IP addresses**. The addresses are gathered from discovery seed addresses that you provided, ping sweep configurations, or neighbor addresses gathered using Auto-DiscoveryRules.

Discovery avoids potential confusion when a device has multiple IP addresses by following these rules:

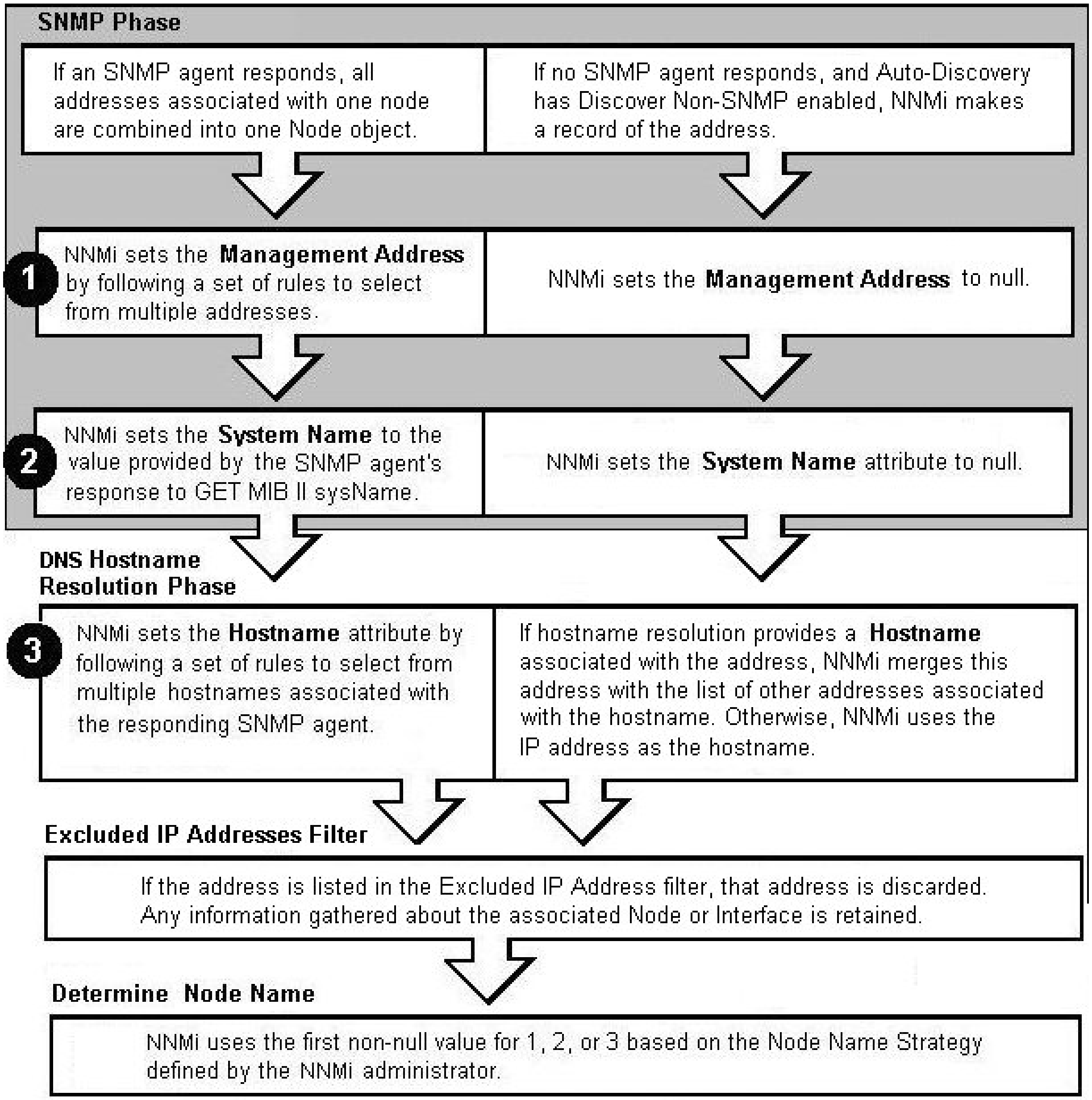
- If the device supports SNMP, the address of the responding SNMP agent is recorded (the Management Address) and the other addresses are associated with the node. See "Specific Node Settings Form (Communication Settings)" on page 155 for more information about configuring the management address.
- If the device does not support SNMP, NNMi queries DNS to determine the hostname. If this hostname matches another non-SNMP node, NNMi merges the information to create only one node with multiple associated addresses.

See "Configure the Node Name Strategy" on page 203 to learn how to configure the NNMi node name strategy.

Node Name Decision Tree

For each discovered address, NNMi gathers multiple attributes that are used to implement your Node Name strategy. NNMi chooses the Node Name based on the Management Address, System Name, and Hostname collected during discovery. The following diagram shows how NNMi determines values for these attributes.

Note: If you change a node's Hostname, there is a delay before NNMi data reflects the name change, because NNMi caches DNS names to enhance performance.



If an SNMP agent responds, all addresses associated with one node are combined into one Node object.

If no SNMP agent responds, and Auto-Discovery has Discover Non-SNMP enabled, NNMi makes a record of the address.

NNMi sets the **Management Address** by following a set of rules to select from multiple addresses.

NNMi sets the **Management Address** to null.

NNMi sets the **System Name** to the value provided by the SNMP agent's response to GET MIB II sysName.

NNMi sets the **System Name** attribute to null.

NNMi sets the **Hostname** attribute by following a set of rules to select from multiple hostnames associated with the responding SNMP agent.

If hostname resolution provides a **Hostname** associated with the address, NNMi merges this address with the list of other addresses associated with the hostname. Otherwise, NNMi uses the IP address as the hostname.

If the address is listed in the Excluded IP Address filter, that address is discarded. Any information gathered about the associated Node or Interface is retained.

NNMi uses the first non-null value for 1, 2, or 3 based on the Node Name Strategy defined by the NNMi administrator.

Ignore Forwarding Database Information from a Node Group

Forwarding Database (FDB) information can cause NNMi to establish wrong Layer 2 Connections in the following cases:

- When the FDB is configured as cache and contains obsolete data.
- In network environments with hardware from a variety of vendors, when each vendor generates different and sometimes conflicting FDB data.

Optional: NNMi administrators can configure Spiral Discovery to ignore the FDB data from one Node Group when calculating Layer 2 Connections (the FDB data is still included in other calculations).

> **Note:** (*NNMi Advanced - Global Network Management feature*) NNMi must read the Forwarding Database (FDB) tables from Ethernet switches within the network before accurate communication paths between these network devices can be calculated. Because FDB data is involved, NNMi can produce different results on a Regional Manager as opposed to the Global Manager.

***Optional*: To configure NNMi to ignore FDB information from devices in one Node Group when calculating Layer-2 Connections:**

1. Navigate to the **Discovery Configuration** form.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Expand **Discovery**.
   c. Select **Discovery Configuration**.
2. Locate the **Layer 2 Connection Source** attribute on the left side of the form.
3. In the **Node Group to disable FDB** drop-down, specify which Node Group:
   - Click the drop-down list's down arrow and choose a previously defined Node Group.
   - Select the Lookup icon and select New to create a new Node Group.
4. Click **Save and Close**. Spiral Discovery implements your changes during the next regularly scheduled discovery interval. To apply the changes immediately, use **Actions** → **Polling** → **Configuration Poll**. See Using Actions to Perform Tasks for more information.

> **Tip:** You can right-click any object in a table or map view to access the **Actions** menu.

Configure Schedule Settings

For details about Spiral Discovery's Schedule Settings, see the following:

Spiral Discovery's Schedule Settings determine how often NNMi requests data and updates information about the devices in your network domain. NNMi requests the following information:

- Information about the nodes, addresses, and interfaces you configure for discovery.
- Information about Level 2 connectivity between interfaces and VLANs in your network.
- Information about Level 3 connectivity between addresses in your network.

Make sure the interval value you choose provides plenty of time so Spiral Discovery cycles do not overlap. The larger your network environment, the longer the time required to complete a Spiral

Discovery cycle. These Schedule Settings might help NNMi administrators meet service-level agreement (SLA) commitments.

Adjust the Rediscovery Interval

When configuring Spiral Discovery, you determine how often network traffic is generated to gather and verify information about your network management domain. This time interval controls how frequently information is gathered about nodes, interfaces, IP addresses, subnets, VLANs, and connections in the network. See "Configure Schedule Settings" on the previous page for more information.

> **Tip:** You can also adjust the Rediscovery Interval for a specified Node Group. See "Adjust the Node Group Rediscovery Interval" on the next page for more information.

To adjust the rediscovery cycle interval:

1. Navigate to the **Discovery Configuration** form.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Expand **Discovery**.
   c. Select **Discovery Configuration**.
2. Locate the **Schedule Settings** tab.
3. In the **Rediscovery Interval** attribute, set the time interval that Spiral Discovery waits between information gathering cycles.

   The default is 24 hours between cycles. The minimum is 1 hour.

   Make sure the interval value provides plenty of time so Spiral Discovery cycles do not overlap. The larger your network environment, the longer the time required to complete a Spiral Discovery cycle.

   > **Note:** During rediscovery, NNMi checks each Node for membership in Node Groups. If the Node belongs to a Node Group that is associated with a Custom Poller Policy, NNMi might issue additional requests for information. See "Create Custom Polling Configurations" on page 419 for more information.

4. Click **Save and Close**. Spiral Discovery implements your changes during the next regularly scheduled discovery interval. To apply the changes immediately, use **Actions** → **Polling** → **Configuration Poll**. See Using Actions to Perform Tasks for more information.

   > **Tip:** You can right-click any object in a table or map view to access the **Actions** menu.

5. *Optional*. To establish the *beginning* of the interval, schedule a task to run the `nnmnoderediscover.ovpl -all` command line tool. Consider choosing a quiet time on your network so traffic generated by NNMi does not disturb regular business.

   The Spiral Discovery cycle start time might change slightly depending on circumstances within your network environment. Use the Nodes (All Attributes) view and sort on the **Last Completed** column (last Discovery cycle) to check recent times.

Related Topics

"Adjust the Node Group Rediscovery Interval" below

Adjust the Node Group Rediscovery Interval

When configuring Spiral Discovery, you determine how often network traffic is generated to gather and verify information about your network management domain. This time interval controls how frequently information is gathered about the nodes, interfaces, IP addresses, subnets, VLANs, and connections in the network for the specified Node Group. See "Adjust the Rediscovery Interval" on the previous page for more information.

There are two benefits to using a Node Group Rediscovery Interval:

- You have many choices about the criteria for defining your Node Group (see "Create Node Groups" on page 295).
- Your Node Group Rediscovery Interval enables a subset of devices to be rediscovered at a different rate than the default Rediscovery Interval. For example, this feature could be useful to configure NNMi to do the following:
  - Help NNMi administrators meet service-level agreement (SLA) commitments.
  - More frequently rediscover device configuration changes for frequently changing devices or your most important devices.
  - Less frequently rediscover unimportant devices in your network domain to minimize network traffic.

To adjust the Node Group rediscovery cycle interval:

1. Navigate to the **Discovery Configuration** form.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Expand **Discovery**.
   c. Select **Discovery Configuration**.
2. Locate the **Schedule Settings** tab.
3. In the **Node Group** attribute, specify the name of the Node Group for which you want to configure the Node Group Rediscovery Interval.
4. In the **Node Group Rediscovery Interval** attribute, set the time interval that Spiral Discovery waits between information gathering cycles.

   The default is 24 hours between cycles. The minimum is 1 hour.

   Make sure the interval value provides plenty of time so Spiral Discovery cycles do not overlap. The larger your network environment, the longer the time required to complete a Spiral Discovery cycle.

   Specify the **Node Group Rediscovery Interval**. If a Node is reconfigured so that one or more attribute values no longer match the specified Node Group's configuration criteria, the next time the Node is discovered, it is removed from the Node Group. NNMi then determines when to rediscover the Node using the **Rediscovery Interval** setting.

For example, if a Node Group is created using `sysName` as an Additional Filter, and the System Name value is changed for a Node, that Node will no longer belong to the Node Group. After the Node is removed from the specified Node Group, NNMi uses the **Rediscovery Interval** setting instead of the **Node Group Rediscovery Interval** setting to determine when to update discovery information for the Node.

> **Note:** During rediscovery, NNMi checks each Node for membership in Node Groups. If the Node belongs to a Node Group that is associated with a Custom Poller Policy, NNMi might issue additional requests for information. See "Create Custom Polling Configurations" on page 419 for more information.

5. Click **Save and Close**. Spiral Discovery implements your changes during the next regularly scheduled discovery interval. To apply the changes immediately, use **Actions** → **Polling** → **Configuration Poll**. See Using Actions to Perform Tasks for more information.

   > **Tip:** You can right-click any object in a table or map view to access the **Actions** menu.

6. *Optional*. To establish the *beginning* of the interval, schedule a task to run the `nnmnoderediscover.ovpl -all` command line tool. Consider choosing a quiet time on your network so traffic generated by NNMi does not disturb regular business.

   The Spiral Discovery cycle start time might change slightly depending on circumstances within your network environment. Use the Nodes (All Attributes) view and sort on the **Last Completed** column (last Discovery cycle) to check recent times.

Related Topics

"Adjust the Rediscovery Interval" on page 210

Configure Whether to Delete Unresponsive Nodes

When configuring Spiral Discovery, you determine whether and how quickly NNMi deletes nodes that are unresponsive.

> **Note:** NNMi does not delete any unresponsive object during the first 24 hours after NNMi is restarted (ovstart). The 24 hour additional wait time ensures that NNMi has an opportunity to poll each Node.

> **Caution:** To understand the results of deleting a Node, see "Delete Nodes" on page 1602 and "Delete One or More Objects" on page 1604.

NNMi automatically deletes an unresponsive node using the following criteria:

- The node does not respond to SNMP requests for the specified number of days.
- All of the node's IP Addresses do not respond to ICMP for the specified number of days.

One of the following Conclusions must be associated with the Node. See the help for Node Form: Conclusions Tab for more information:

- `NodeUnmanageable`
- `NonSNMPNodeUnmanageable`
- `NodeDown`
- `NodeOrConnectionDown`

To configure NNMi to automatically delete Unresponsive Objects:

1. Navigate to the **Discovery Configuration** form.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Expand **Discovery**.
   c. Select **Discovery Configuration**.
2. Navigate to the **Schedule Settings** tab.
3. In the **Period (in Days) to Delete Unresponsive Nodes** attribute, set the number of days that a Node must be unresponsive before NNMi deletes the node and all nodes in its shadow from the NNMi database (as well as each Node's history and related objects). For more information about nodes in the shadow, see Node Down.

   0 (zero, the default value) = Do not delete from the NNMi database.

   Any number provided represents the number of days that the object must remain unresponsive.
4. Click **Save and Close**. Spiral Discovery implements your changes during the next regularly scheduled discovery interval. To apply the changes immediately, use **Actions** → **Polling** → **Configuration Poll**. See Using Actions to Perform Tasks for more information.

> **Tip:** You can right-click any object in a table or map view to access the **Actions** menu.

Configure Whether to Delete Layer 2 Connections

When configuring Spiral Discovery, you determine whether and how frequently NNMi deletes connections that are down.

NNMi deletes connections once per day (1 a.m. by default).

NNMi automatically deletes any Layer 2 Connections that are Down using the following criteria:

- The `ConnectionDown` Conclusion must be associated with the connection for the specified number of days. See Layer 2 Connection Form: Conclusions Tab for more information.
- When interfaces are participating in **Link Aggregation**[1] protocols, NNMi automatically deletes

[1] Protocols used on Switches to configure multiple Interfaces (Aggregation Member Interfaces) to function as if they were one (an Aggregator Interface). When two Aggregator Interfaces establish a connection, that connection is an Aggregator Layer 2 Connection. The Aggregator Layer 2 Connection appears on Layer 2 Neighbor View maps as a thick line with an Interface icon at each end (representing the Aggregator Interface). When you double-click the thick line, it converts into multiple thin lines representing the participating Aggregation Member Layer 2 Connections with their Aggregation Member Interfaces at each end of the lines.

Aggregation Member Layer 2 Connections that have the `ConnectionDown` Conclusion for the specified number of days.

> **Note:** During the next Rediscovery cycle, NNMi deletes any *Aggregator Layer 2 Connections* without any Aggregation Member Layer 2 Connections.

- When the Layer 2 Connection object's **Topology Source** value is one of the following, NNMi *never* automatically deletes the connection (see the Help topic for Layer 2 Connection Form for more information):

  ROUTES - indicates NNMi creates the connection from the routing data. NNMi creates these Layer 2 Connections for *unnumbered* interfaces. For more information, see the "NNMi Discovery" chapter of the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.

  SUBNETCONNECTION- Subnet Connection Rule. NNMi applied a special configurable rule for subnets (only those IPv4 subnets with a prefix length between 28 and 31) to detect this connection. NNMi gathers information from Layer 3 of the Open System Interconnection (OSI) networking model to detect this connection. Layer 3 is the Network layer that provides switching, routing, and logical paths (virtual circuits) for transmitting data between nodes. The NNMi administrator configures the Subnet Connection Rules, see "Help for Administrators" for more information. On the NNMi map, the following icon is in the middle of the SUBNETCONNECTION line:
  .

  USER - This connection was configured by your NNMi administrator (using the Connection Editor). See "Help for Administrators" for more information.

To configure NNMi to automatically delete down Layer 2 Connections:

1. Navigate to the **Discovery Configuration** form.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Expand **Discovery**.
   c. Select **Discovery Configuration**.
2. Navigate to the **Schedule Settings** tab.
3. In the **Period (in Days) to Delete Connections that are Down** attribute, set the number of days that a Connection must be down before NNMi deletes the connection.

   0 (the default value) = Do not delete from the NNMi database.

   Any number provided represents the number of days that the object must remain unresponsive.
4. Click **Save and Close**. Spiral Discovery implements your changes during the next regularly scheduled discovery interval. To apply the changes immediately, use **Actions** → **Polling** → **Configuration Poll**. See Using Actions to Perform Tasks for more information.

   > **Tip:** You can right-click any object in a table or map view to access the **Actions** menu.

> **Tip:** To confirm that NNMi is successfully automatically deleting Layer 2 Connections, look for

the following message in the `nnm.*.*.log` file:

```
One connection with name <ConnectionName> has been deleted, because
it has been down for <N> days with StatusConclusion ConnectionDown.
```

(See the "NNMi Logging" chapter in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`)

Layer 2 Connections can be deleted manually, see "Delete One or More Objects" on page 1604.

Configure Auto-Discovery Rules

Auto-Discovery Rule configuration settings control Auto-Discovery behavior within the *Default Tenant*:

Auto-Discovery extends discovery by gathering Hints about additional devices:

- Auto-Discovery gathers information about neighboring devices using ARP cache, DNS, and the following protocols:
  - **BGP** — Border Gateway Protocol
  - **EIGRP** — Cisco Enhanced Interior Gateway Routing Protocol
  - **OSPF** — Open Shortest Path First
  - And data gathered from a variety of Layer 2 *discovery protocols*. See the list of Topology Source protocols in Layer 2 Connection Form.
- Auto-Discovery monitors SNMP traps from previously discovered IP addresses for additional information.

  Auto-Discovery also uses the source IP address from SNMP traps as Discovery Hints for new addresses. If your Auto-Discovery Rules' IP Ranges include that new IP address, NNMi uses the Trap Hint for initial discovery of that address. NNMi then requests the Node's current Management Address (the address from which the node's SNMP Agent responds) and uses that IP address for all further communication. NNMi calculates whether the new address belongs to a previously discovered Node or a new Node.
- Auto-Discovery gathers information about neighbors adjacent to each discovered device. Auto-Discovery then discovers those neighbors and repeats the process. This sequence continues until the Default Tenant's boundaries are reached (identified by Auto-Discovery Rules' IP Address Ranges or Ordering numbers).

> **Note:** NNMi never gathers Auto-Discovery *Hints* from IP addresses assigned to a Tenant other than the Default Tenant.

Before you start, have a clear idea of what you want to accomplish, see "Example Uses of Auto-Discovery" on page 228.

If you do not configure any Auto-Discovery Rules, Spiral Discovery only finds the configured Discovery Seeds (see "Specify Discovery Seeds" on page 256 for more information).

> **Note:** When any Node is discovered because of an Auto-Discovery Rule, NNMi assigns that Node to the *Default Tenant* (and whichever Security Group attribute value is currently configured as Default Tenant's *Initial Discovery Security Group*). See "Configure Tenants" on page 194 and "About Security Groups" on page 515 for more information.

Auto-Discovery Rule Configuration Tasks

Task	How
"Configure Basic Settings for the Auto-Discovery Rule" on page 218	Provide the basic requirements for an Auto-Discovery Rule configuration: • The name of the rule. • Specify the order in which Auto-Discovery applies this rule within the Default Tenant. • Specify how ICMP and SNMP protocols are used for this segment of discovery. • Designate whether devices identified by this rule are *Discovered* or *Rejected* during the Auto-Discovery process. See "Auto-Discovery Rule Behavior Choices" on the next page.
Rule Criterion	"IP Address Ranges for the Auto-Discovery Rule" on page 221 Use IP addresses with wildcards to specify the area within Default Tenant that this Auto-Discovery Rule controls. You decide whether Ping Sweep is used for this segment of discovery.
	"SNMP System Object ID Ranges for the Auto-Discovery Rule" on page 225 Use industry standard System Object IDs to control Auto-Discovery within the Default Tenant. Use the **Configuration** → **Device Profiles** view to see the list of all known system object IDs (MIBII `sysObjectID`) at the time NNMi released. This list of system object IDs is useful to expand or limit the range of devices that Auto-Discovery finds.

Auto-Discovery Rule Behavior Choices

Auto-Discovery Rules control the extent of automatic discovery within the *Default Tenant*. Specify what Auto-Discovery should reject or find within your network environment by defining at least two Auto-Discovery Rules. You assign an Ordering number to each rule. For each discovered Node, Interface, or IP address, NNMi applies the first *matching* rule from lowest to highest Ordering number.

> **Tip:** Give your Reject rule a lower Ordering number than the Include rule or rules to which it applies.

Purpose = Reject Matching Nodes

Selections	Behavior
Discover Matching Nodes □ **Discover Any SNMP Device** □ **Discover Non-SNMP Devices** □	Auto-Discovery rejects the following within Default Tenant (does not add any information to the NNMi database, does not query for information or Hints about neighboring devices): • All addresses specified in the rule's IP Ranges table (if any) • All devices that meet the criteria specified in the rule's System Object ID Ranges table (if any) - based on RFC 1213, MIB-II `sysObjectID` values **Caution:** If *both ranges are empty*, this rule would cause Auto-Discovery to never discover anything specified in all rules with higher Ordering numbers.

The following table shows the choices for instructing Auto-Discovery to discover Nodes.

> **Note:** Configure at least one Auto-Discovery Rule from the following table. And at least one Auto-Discovery Rule from the following table must specify the IP Address Range within which you want to use Auto-Discovery in Default Tenant.

Purpose = Discover Matching Nodes

Selections	Behavior
Discover Matching Nodes ☑ **Discover Any SNMP Device** □ **Discover Non-SNMP Devices** □	Auto-Discovery finds the following *Routers and Switches* within Default Tenant: • All must have IP addresses within the ranges specified in the rule's IP Ranges table (if any) • All must meet the criteria specified in the rule's System Object ID Ranges table (if any) - based on RFC 1213, MIB-II `sysObjectID` values

Purpose = Discover Matching Nodes, continued

Selections	Behavior
Discover Matching Nodes ☑ **Discover Any SNMP Device** ☑ **Discover Non-SNMP Devices** □	Auto-Discovery finds the following devices within Default Tenant: • All must have IP addresses within the ranges specified in the rule's IP Ranges table (if any) • Any that answer SNMP queries *and* meet the criteria specified in the rule's System Object ID Ranges table (if any) - based on RFC 1213, MIB-II `sysObjectID` values
Discover Matching Nodes ☑ **Discover Any SNMP Device** ☑ **Discover Non-SNMP Devices** ☑	Auto-Discovery finds the following devices within Default Tenant: • All must have IP addresses within the ranges specified in the rule's IP Ranges table (if any) • Any that answer SNMP queries *and* meet the criteria specified in the rule's System Object ID Ranges table (if any) - based on RFC 1213, MIB-II `sysObjectID` values • Any that answer ICMP queries but not SNMP queries
Discover Matching Nodes ☑ **Discover Any SNMP Device** □ **Discover Non-SNMP Devices** ☑	Auto-Discovery finds the following devices within Default Tenant: • All must have IP addresses within the ranges specified in the rule's IP Ranges table (if any) • Any *Routers and Switches* that answer SNMP queries *and* meet the criteria specified in the rule's System Object ID Ranges table (if any) - based on RFC 1213, MIB-II `sysObjectID` values • Any devices that answer ICMP queries but not SNMP queries

Configure Basic Settings for the Auto-Discovery Rule

***Default Tenant* only**: These Auto-Discovery Rule settings determine which methods Auto-Discovery applies when discovering nodes within your Default Tenant.

To configure this Auto-Discovery Rule for the Default Tenant:

1. Navigate to the **Auto-Discovery Rule** form.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Expand **Discovery**.
   c. Select **Discovery Configuration**.

d. Locate the **Auto-Discovery Rules** tab.

e. Do one of the following:

- To establish a rule, click the ✱ New icon, and continue.
- To edit a rule, double-click the row representing the configuration you want to edit, and continue.
- To delete a rule, select a row, and click the ✖ Delete icon.

2. Provide the required basic settings for this Auto-Discovery Rule (see the Basics for this Auto-Discovery Rule table).

3. Determine the Auto-Discovery Rule's behavior (see "Auto-Discovery Rule Behavior Choices" on page 217):Basics for this Auto-Discovery Rule

   - Purpose of this Auto-Discovery Rule
   - Extend Default Behavior (beyond Routers and Switches)

4. There are many ways to implement discovery. Before you start this step, see "Example Uses of Auto-Discovery" on page 228.

   Configure one or more ranges, to identify the devices you want to discover or reject.

   - "IP Address Ranges for the Auto-Discovery Rule" on page 221
   - "SNMP System Object ID Ranges for the Auto-Discovery Rule" on page 225

5. Click **Save and Close** to return to the **Discovery Configuration** form.

6. Click **Save and Close**. Spiral Discovery implements your changes during the next regularly scheduled discovery interval.

7. *Optional:* Open the **Discovery Configuration** workspace again and provide a discovery seed for each address range of this Auto-Discovery Rule. Core routers make the best Auto-Discovery seeds. See "Specify Discovery Seeds" on page 256.

Basics for this Auto-Discovery Rule

Task	How
Name	Give this Auto-Discovery Rule a meaningful name.
Ordering	Determine the order in which the Auto-Discovery Rules are applied. No duplicate Ordering numbers are permitted. Each Auto-Discovery Rule ordering number must be unique. **Tip:** Consider incrementing Ordering numbers by 10s or 100s to provide flexibility when adding new rules over time. **IP address ranges**: If a device falls within two Auto-Discovery Rules, the Auto-Discovery Rule with the lowest ordering number applies. For example, if an Auto-Discovery Rule includes certain IP addresses, then no other Auto-Discovery Rules with higher ordering numbers apply to those addresses. **System Object ID ranges**:

Basics for this Auto-Discovery Rule, continued

Task	How
	• If no IP address range is included in this Auto-Discovery Rule, then the system object ID settings take precedence over all Auto-Discovery Rules that have higher Ordering numbers than this Auto-Discovery Rule. • If an IP address range is included in this Auto-Discovery Rule, your system object ID range applies only within this Auto-Discovery Rule.
Notes	Provide any additional useful information about this Auto-Discovery Rule. Type a maximum of 1024 characters. Alpha-numeric, spaces, and special characters (~ ! @ # $ % ^ & * ( ) _+ -) are permitted.

Purpose of this Auto-Discovery Rule

Task	How
Discover Matching Nodes	If ☑ enabled, Auto-Discovery gathers information about neighboring devices and adds devices to the NNMi database if those device meet the rule's criteria. For more information see "Which Nodes Are Discovered?" on page 176. **Note:** By default NNMi discovers routers and switches. You can expand the number of device types that NNMi discovers by enabling ☑ **Discover Any SNMP Device** and including one or more System Object ID Ranges (based on MIB-II `sysObjectID` values). Your address ranges and system object ID ranges determine which discovered addresses are added to the NNMi database. If □ disabled, Auto-Discovery rejects devices that match this rule unless: • The device's address is a discovery seed. See "Specify Discovery Seeds" on page 256 to learn how to establish discovery seeds. • The device's address is reported as a neighbor to another discovered address. If you want to ensure that an address is never added to the NNMi database, see "Configure an Excluded IP Addresses Filter" on page 248 or "Configure an Excluded Interfaces Filter" on page 254 settings.

Extend Default Behavior (beyond Routers and Switches) for this Auto-Discovery Rule

Task	How
Discover Any SNMP Device	**Note:** This attribute is ignored if Discover Matching Nodes is unchecked. However, if you configure any device as a discovery seed, discovery always adds that device to the database. If ☑ enabled, Auto-Discovery gathers information about any device that responds to SNMP queries (in addition to routers or switches that are

Extend Default Behavior (beyond Routers and Switches) for this Auto-Discovery Rule, continued

Task	How
	discovered by default). These nodes appear on maps and are monitored. If □ disabled, Auto-Discovery rejects all device types except routers, switches, discovery seeds, and device types specified in your system object ID ranges. (Routers and switches are identified by the settings in the device profile.)
Discover Non-SNMP Devices	**Note:** This attribute is ignored if Discover Matching Nodes is unchecked. However, if you configure any device as a discovery seed, discovery always adds that device to the database. Non-SNMP devices are those that do not respond to SNMP queries. If you enable **Discover Non-SNMP Devices**, note the following: • If you do not want NNMi to discover every node in your network, make sure your Auto-Discovery Rules correctly limit the scope of the discovery. See "Example Uses of Auto-Discovery" on page 228 for more information. • Selecting this option might cause you to reach your licensed capacity very quickly. See "Extend a Licensed Capacity" on page 1575. • If NNMi determines that a non-SNMP node has a hostname matching another non-SNMP node, NNMi merges the information to create only one node and includes any additional IP address information under the same node. Non-SNMP nodes might be inaccurately represented under the following circumstances: ▪ One or more non-SNMP nodes in your network use the same hostname. ▪ The same non-SNMP node has multiple hostnames. ▪ A non-SNMP node name changes (see "Delete Nodes" on page 1602). If ☑ enabled, Auto-Discovery adds to the database any addresses that do not respond to SNMP queries. If □ disabled, Auto-Discovery rejects any address that does not respond to SNMP queries.

IP Address Ranges for the Auto-Discovery Rule

***Default Tenant* only**: Auto-Discovery IP address ranges determine the outer limits for the area controlled by the current Auto-Discovery Rule. You can create multiple IP ranges within one Auto-Discovery Rule (order *within the rule* does not matter). Before you start, have a clear idea of what you want to accomplish, see "Auto-Discovery Rule Behavior Choices" on page 217 and "Example Uses of Auto-Discovery" on page 228.

If the Auto-Discovery Rule's **Discover Matching Nodes** □ is disabled, click here for additional information.

- Auto-Discovery *does not gather neighbor information* from the addresses identified in any IP address range included in this rule. The addresses, themselves, might still show up in the topology database.

> **Note:** Neighbor information is still gathered from IP addresses specifically identified in the discovery seeds configuration settings.
>
> NNMi also uses the source IP address from SNMP traps as hints to discovery. NNMi uses those hint IP address only for initial discovery. NNMi then requests the current Management Address (the address from which the node's SNMP Agent responds) and uses that IP address for all communication.

If the Auto-Discovery Rule's **Discover Matching Nodes** ☑ is enabled, click here for additional information.

- At least one of your Auto-Discovery Rules must have an IP address range designated as an **Include in rule** range type. Auto-Discovery *gathers neighbor information* from those addresses to extend discovery.
- *Optional*. You can configure NNMi to ignore subsets of those IP addresses (an **Ignored by rule** range, which means that those addresses are available for other Auto-Discovery Rules with higher Ordering numbers).
- *Optional*. Specify system object ID (MIB-II `sysObjectID`) ranges to be included or ignored. This technique constricts or extends the types of devices affected by this rule. See "SNMP System Object ID Ranges for the Auto-Discovery Rule" on page 225 for more information.

NNMi discovers any devices that comply with your rule configurations, and creates a record of each device in the NNMi database. If the device supports SNMP, all addresses for that device are combined into one Node object. If the device does not support SNMP, NNMi queries DNS to determine the hostname. If this hostname matches another non-SNMP node, NNMi merges the information to create only one node with multiple associated addresses.

To specify an Auto-Discovery Rule IP address range for Default Tenant:

1. Navigate to the **Auto-Discovery** form.
   a. In the **Workspace** navigation panel, open the 🔧 **Configuration** workspace.
   b. Select **Discovery Configuration**.
   c. Select the **Auto-Discovery Rule** tab, and do one of the following:
      - To establish an Auto-Discovery Rule, click the ✳ New icon.
      - To edit an Auto-Discovery Rule, click the 📂 Open icon in the row representing the configuration you want to edit.
2. Provide the Basic Settings, see "Configure Basic Settings for the Auto-Discovery Rule" on page 218.
3. Navigate to the **IP Ranges** tab.

4. *Optional*. Decide if you want to use Ping Sweep in this segment of network discovery.

   IPv4 addresses only: In Wide Area Networks (WANs) such as ATM, Frame Relay, and Point-to-Point (where ARP cache is not available), the Ping Sweep locates nodes for Auto-Discovery to use when gathering neighbor information and evaluating connections between Nodes.

   > **Note:** Ping Sweep works only with IPv4 addresses. All nodes discovered using Auto-Discovery are assigned to the *Default Tenant*.

   - **Enable Ping Sweep** ☑
     Auto-Discovery can issue a wide range of ICMP ping commands. For details, see "Ping Sweep for Auto-Discovery in Default Tenant" on page 242. NNMi only uses Ping Sweep across a maximum of the last two octets (/16) of the network specified by each IPv4 IP address range.

     If things do not work as expected, check whether Ping Sweep is disabled. See "Configure Ping Sweep (override for all Auto-Discovery Rules)" on page 202. Also verify that ICMP communication is enabled, see "Communication Region Form" on page 137 and "Specific Node Settings Form (Communication Settings)" on page 155.

   - **Enable Ping Sweep** □
     Auto-Discovery depends on Discovery Seeds as starting points. For details, see "Discovery Seeds for Auto-Discovery in Default Tenant" on page 241 for important information.

5. *Optional*. To provide an IP address range for this Auto-Discovery Rule, do one of the following:

   - To create an IP range, click the ✳ New icon, and continue.

   - To edit an IP range, click the 📂 Open icon in the row representing the configuration you want to edit, and continue.

   - To delete an IP range, select a row, and click the ✖ Delete icon.

6. Define one or more IP address ranges for this Auto-Discovery Rule, the order of ranges defined *within this rule* does not matter (see table).

7. Click 💾 **Save and Close** to return to the **Auto-Discovery Rule** form.

8. Click 💾 **Save and Close** to return to the **Discovery Configuration** form.

9. Click 💾 **Save and Close.** If you enabled Ping Sweep for this Auto-Discovery Rule, NNMi issues the Ping Sweep when you click Save and Close. Otherwise, Spiral Discovery implements your changes during the next regularly scheduled discovery interval.

Discovery IP Range Form

Name	Description
IP Range	**Note:** If you enter an IP address value that represents only one IP address, Auto-Discovery gathers neighbor information only from the address you enter. (Discovery extends only one hop out from this address.) To specify a range of IP addresses for this Auto-Discovery Rule, use one of the

Discovery IP Range Form, continued

Name	Description
	following. Pick one address notation style. Combinations of wildcards and CIDR notation are not permitted within one address range. You can provide multiple address range settings:

- **IPv4 address wildcard notation**.

  An IPv4 Address range is a modified dotted-notation where each octet is one of the following:

  - A specific octet value between 0 and 255
  - A low-high range specification for the octet value (for example, "112-119")
  - An asterisk (*) wildcard character, which is equivalent to the range expression "0-255"

  > **Note:** The following two IPv4 addresses are considered invalid: `0.0.0.0` and `127.0.0.0`

  Examples of valid IPv4 address wildcards include:

  ```
  10.1.1.*
  10.*.*.*
  10.1.1.1-99
  10.10.50-55.*
  10.22.*.4 10.1-9.1-9.1-9
  ```

- **IPv4 Classless Inter-Domain Routing (CIDR) notation**.

  The CIDR notation specifies the number of consecutive bits in the IPv4 address that must match.

  For example, `10.2.120.0/21`

  > **Note:** NNMi does not support CIDR subnet mask notation such as, `10.2.120.0/255.255.248.0`

Example IPv4 Prefix Length Values	Number of Usable IPv4 Addresses
28	14 (16-2=14)*
29	6 (8-2=6)*
30	2 (4-2=2)*
31	2

  * Two IPv4 addresses are reserved in each subnet. The first IPv4 address is used for the network itself and the last IPv4 address is reserved for broadcast.

- **IPv6 address wildcard notation**

Discovery IP Range Form, continued

Name	Description

	Separate each 16-bit value of the IPv6 address with a colon. The 16-bit value can be any of the following:
	▪ A specific hexadecimal value between `0` and `FFFF` (case insensitive).
	▪ A low-high range specification of the hexadecimal value (for example, `1-1fe`).
	▪ An asterisk (`*`) wildcard character (equivalent to the range expression `0-ffff`).
	Note: The standard IPv6 short-hand notation (`::`) is allowed to express one or more 16-bit elements of zero (`0`) values. However, the mixed IPv6/IPv4 dot-notation (for example, `2001:d88::1.2.3.4`) is not permitted as an IPv6 address range.
	Valid examples of ranges in modified IPv6 address notation include the following:
	`2001:D88:0:A00-AFF:*:*:*:*`
	`2001:D88:1:*:*:*:*:*`
	`2001:D88:2:0:a07:ffff:0a01:3200-37ff`
	• **IPv6 Classless Inter-Domain Routing (CIDR) notation**
	The CIDR notation specifies the number of consecutive bits in the IPv6 address that must match.
	`2001:d88:a00::/44` (equivalent to modified IPv6 address notation `2001:d88:a00-a0f:*:*:*:*:*`)
	For example, valid IPv6 address ranges in CIDR notation include the following:
	`2001:d88:0:a00::/56` (equivalent to modified IPv6 address notation `2001:D88:0:A00-AFF:*:*:*:*`)
	`2001:d88:1::/48` (equivalent to modified IPv6 address notation `2001:D88:1:*:*:*:*:*`)

Range Type	**Include in rule** - The current Auto-Discovery Rule's settings apply to the addresses in this range. **Ignored by rule** - The current Auto-Discovery Rule's settings do not apply to the addresses in this range. Use the **Ignored by rule** setting to identify a subset of addresses within a larger range. The addresses in the ignored range are available to conform to an Auto-Discovery Rule with a higher ordering number.

SNMP System Object ID Ranges for the Auto-Discovery Rule

Vendors are assigned a system object ID number (RFC 1213 MIB-II `sysObjectID`) for each network device they manufacture. This system object ID number is unique for each combination of vendor, device type, and model number (vendor/make/model). For example, all Cisco 6509 routers have the same system object ID.

> **Tip:** See "Configure Device Profiles" on page 292 for more information about system object IDs. In the **Configuration** > **Device Profiles** view , you can quickly and easily locate the system object IDs of devices in your network environment.

***Default Tenant* only**: System object ID ranges are powerful tools for limiting this Auto-Discovery Rule's behavior. For example, limit this rule by excluding specific models of routers and switches. Before you start, have a clear idea of what you want to accomplish, see "Example Uses of Auto-Discovery" on page 228.

When using system object ID ranges for this Auto-Discovery Rule, note the following:

- The rule applies only to the Default Tenant.
- If no IP Address Ranges are defined within this Auto-Discovery Rule, your System Object ID Ranges affect *all* Auto-Discovery Rules that have higher Ordering numbers than this Auto-Discovery Rule.
- If one or more IP Address Ranges are defined within this Auto-Discovery Rule, your System Object ID Ranges affect *only* the current Auto-Discovery Rule.

The following table includes examples of how you might want to expand or limit Auto-Discovery within the Default Tenant using System Object ID Ranges.

Controlling Auto-Discovery within the Default Tenant using System Object ID Ranges

Task	Related Topics
Exclude certain vendor/make/models of Routers and Switches from Auto-Discovery.	See the **Auto-Discovery Rule = Included** information in "Only Routers and Switches Discovered" on page 231 topics.
Expand Auto-Discovery to include device types in addition to routers and switches.	See the **Auto-Discovery Rule = Included** information in "Only Specific Vendor/Make/Models Discovered" on page 234 topics.
Exclude one or more specific device types from all Auto-Discovery rules.	See the **Auto-Discovery Rule = Rejects** information in "Strategies to Exclude Certain Nodes from Auto-Discovery" on page 237.

To specify a system object ID range:

1. Complete all prerequisites. See "Prerequisites for Discovery" on page 187, .
2. Navigate to the **Discovery System Object ID Range** form.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Expand **Discovery**.
   c. Select **Discovery Configuration**.
   d. Select the **Auto-Discovery Rule** tab.
   e. Do one of the following:

   - To create an Auto-Discovery Rule, click the New icon.
   - To edit an Auto-Discovery Rule, double-click the row representing the configuration you want to edit.

f. In the **Auto-Discovery Rule** form, select the **System Object ID Ranges** tab.

g. Do one of the following:

   - To create a system object ID range, click the New icon, and continue.
   - To edit a system object ID range, click the Open icon in the row representing the configuration you want to edit, and continue.
   - To delete a system object ID range, click the Delete icon.

3. Provide one or more System Object ID ranges for this Auto-Discovery Rule, the order of ranges defined *within this rule* does not matter (see the table).
4. Click **Save and Close** to return to the **Auto-Discovery Rule** form.
5. *Optional*. Provide IP Address Ranges to limit the scope of this Auto-Discovery Rule (see "IP Address Ranges for the Auto-Discovery Rule" on page 221).
6. Click **Save and Close** to return to the **Discovery Configuration** form.
7. Click **Save and Close**. Spiral Discovery implements your changes during the next regularly scheduled discovery interval.

Discovery System Object ID Range Definition

Attribute	Description
System Object ID Prefix	Enter a prefix of an SNMP system object ID, or enter the entire SNMP system object ID. A partial entry becomes a wildcard. For example, if you enter 1.3.6.1.4.1.11, discovery finds all HP devices. If you enter 1.3.6.1.4.1.9, discovery finds all Cisco devices. **Note:** Do not use dashes or asterisks (*) in your system object ID value. Do not use a period (.) as the first character. A partial entry becomes a wildcard.
Range Type	**Include in rule** - Instructs Auto-Discovery to discover devices matching this system object ID range. **Ignored by rule** - Instructs Auto-Discovery to ignore devices matching this system object ID range. The `sysObjectIDs` in the ignored range are available to conform to an Auto-Discovery Rule with a higher ordering number.
Notes	Add any information about this rule that would be useful to you and your team. Type a maximum of 1024 characters. Alpha-numeric, spaces, and special characters (~ ! @ # $ % ^ & * ( ) _+ -) are permitted.

Example Uses of Auto-Discovery

Review the following examples to learn how to use Auto-Discovery Rules within the Default Tenant:

Set Outside Limits for Auto-Discovery

***Default Tenant* only**: Best practice is to create a pair of Auto-Discovery Rules with carefully chosen Ordering numbers to clearly identify the entire group of IP addresses within your network management domain and the devices you care about. You can add, remove, or change the settings in this pair of Auto-Discovery Rules at any time.

Define a pair of Auto-Discovery rules as described in the following table. For ideas about how to use this pair of rules:

Configure Outside Limits for Auto-Discovery

Task	How
Auto-Discovery Rule = Included: Create an Auto-Discovery Rule that specifies one or more IP address ranges to identify the outer limits of your management domain. It is recommended that you use your second-lowest **Ordering** number. This ensures that Auto-Discovery never uses addresses outside the specified range or ranges as discovery Hints.	Use the following settings 1. **Name**`Included-IP-Ranges` (for example) 2. **Ordering** 200 3. Specify the techniques Auto-Discovery uses for finding devices: ▪ **Discover Matching Nodes** ☑ **Tip:** With this setting, Auto-Discovery finds only routers and switches. ▪ **Discover Any SNMP Device** □ or ☑

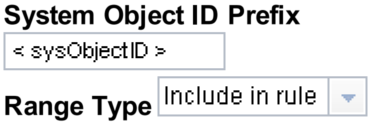
Configure Outside Limits for Auto-Discovery, continued

Task	How
	This setting expands Auto-Discovery to include any device that answers an SNMP query. - **Discover Non-SNMP Devices** □ or ☑ *Optional:* If enabled, Auto-Discovery uses other protocols to detect devices. For details, see "How Spiral Discovery Works" on page 176 and "What Information Is Collected?" on page 177. 4. Create any number of ranges to specify the area within Default Tenant that this Auto-Discovery Rule controls: **IP Range** < IPv4 / IPv6 range> (Minimum: One is required in one of your Auto-Discovery Rules.) **Range Type** Include in rule ▾ 5. *Optional:* If you want to limit Auto-Discovery to only certain vendor/make/models, create one or more System Object ID Ranges. **Caution:** If you use this setting, NNMi ignores all other devices. If it would be easier to reject a small list of System Object IDs (rather than list all the ones you want NNMi to discover), skip this step and see the *Auto-Discovery Rule = Rejects* configuration. **Note:** Do not use dashes or asterisks (*) in your system object ID value. Do not use a period (.) as the first character. A partial entry becomes a wildcard. **SyObjID Range** < sysObjectID > **Range Type** Include in rule ▾ For more information, see "SNMP System Object ID Ranges for the Auto-Discovery Rule" on page 225.

Configure Outside Limits for Auto-Discovery, continued

Task	How
Auto-Discovery Rule = Rejects: Create a second Auto-Discovery Rule that uses IP Address Ranges, System Object ID Ranges, or both to instruct NNMi to reject a subset of the criterion defined in the Auto-Discovery Rule = Included configuration.	Use the following settings (* = required setting): 1. **Name**`Rejected-IPs-sysObjectIDs` (for example) 2. **Ordering** 100 3. Disable all the following settings to instruct Auto-Discovery to gather data about the Ranges identified in the following steps, but then reject that data (do not add it to the NNMi database nor gather Discovery Hints from within the range). ▪ * **Discover Matching Nodes** ☐ ▪ * **Discover Any SNMP Device** ☐ ▪ * **Discover Non-SNMP Devices** ☐ 4. *Optional:* Create any number of: **IP Range** < IPv4 / IPv6 range> * **Range Type** Ignored by rule ▾ **Caution:** These settings instruct Auto-Discovery to not add the specified IP addresses to the NNMi database, not acknowledge any Hints received about them, nor gather Discovery Hints from them unless the address is a discovery seed. See "Specify Discovery Seeds" on page 256 to learn how to establish discovery seeds. If you want to prevent Auto-Discovery from generating *any* requests for data to certain addresses, see "Configuring Communication Protocol" on page 119. 5. *Optional:* If you want to limit Auto-Discovery to only certain vendor/make/models, create System Object ID Ranges. Once you create a System Object ID Range here, Auto-Discovery rejects any devices that meet this criteria.

Configure Outside Limits for Auto-Discovery, continued

Task	How
	Caution: If it would be easier to specify a small list of System Object IDs that should be included (rather than list all the ones you do not want Auto-Discovery to find), skip this step and see the Auto-Discovery Rule = Included configuration instructions. **Note:** Do not use dashes or asterisks (*) in your system object ID value. Do not use a period (.) as the first character. A partial entry becomes a wildcard.  **System Object ID Prefix** < sysObjectID > **Range Type** Include in rule For more information, see "SNMP System Object ID Ranges for the Auto-Discovery Rule" on page 225.
You can create additional rules for fine tuning Auto-Discovery behavior.	**Note:** The pair of rules (Auto-Discovery Rule = Included and Auto-Discovery Rule = Rejects) can potentially cover all requirements. Carefully choose the Ordering Number for any additional Auto-Discovery Rule. Auto-Discovery Rules affect all rules with a higher ordering number.

Only Routers and Switches Discovered

***Default Tenant* only**: If you want Auto-Discovery to automatically find only routers and switches within Default Tenant, use these guidelines.

> **Note:** After you set your configuration according to these guidelines, when a new router or switch is added to your network, you do not need to do anything. NNMi discovers it during the next discovery cycle.

Follow the instructions in "Set Outside Limits for Auto-Discovery" on page 228 and use the following choices in the appropriate steps:

Only Routers and Switches Discovered

Task	How
Auto-Discovery Rule = Included	• **Discover Matching Nodes** ☑ • **Discover Any SNMP Device** □ • **Discover Non-SNMP Devices** □ If you want Auto-Discovery to find all routers and switches. Do not create any **System Object ID Ranges**. *Optional:* If you want to limit Auto-Discovery to only the vendor/make/models of routers and switches that you specify, do the easiest one of the following (for more information see "SNMP System Object ID Ranges for the Auto-Discovery Rule" on page 225): **Note:** Do not use dashes or asterisks (*) in your system object ID value. Do not use a period (.) as the first character. A partial entry becomes a wildcard. • Create one or more **System Object ID Ranges**. Your list *must include everything* you want Auto-Discovery to find. • Do nothing here but make changes to the **System Object ID Ranges** in the *Auto-Discovery Rule = Rejects* configuration. • Use a combination such as: *Auto-Discovery Rule = Included* configuration: Included = `1.3.6.1.4.1.11` (`HP`) *Auto-Discovery Rule = Rejects* configuration: `1.3.6.1.4.1.11.2.3.7.1.10` (`hpnetSwitch200`) `1.3.6.1.4.1.11.2.3.7.2.2` (`hpicfRouterTR`)
Auto-Discovery Rule = Rejects	*Optional:* Create one or more **System Object ID Ranges** that identify the vendor/make/models of routers and switches you do not want Auto-Discovery to find. **Note:** Do not use dashes or asterisks (*) in your system object ID value. Do not use a period (.) as the first character. A partial entry becomes a wildcard. For example, `hpnetSwitch200` and `hpicfRouterTR`: • **System Object ID Prefix** `1.3.6.1.4.1.11.2.3.7.1.10` **Range Type** Include in rule ▾ • **System Object ID Prefix** `1.3.6.1.4.1.11.2.3.7.2.2` **Range Type** Include in rule ▾

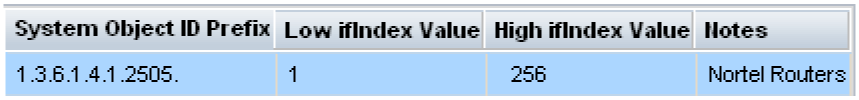
For additional Auto-Discovery ideas:

Only Routers' Physical Interfaces Discovered

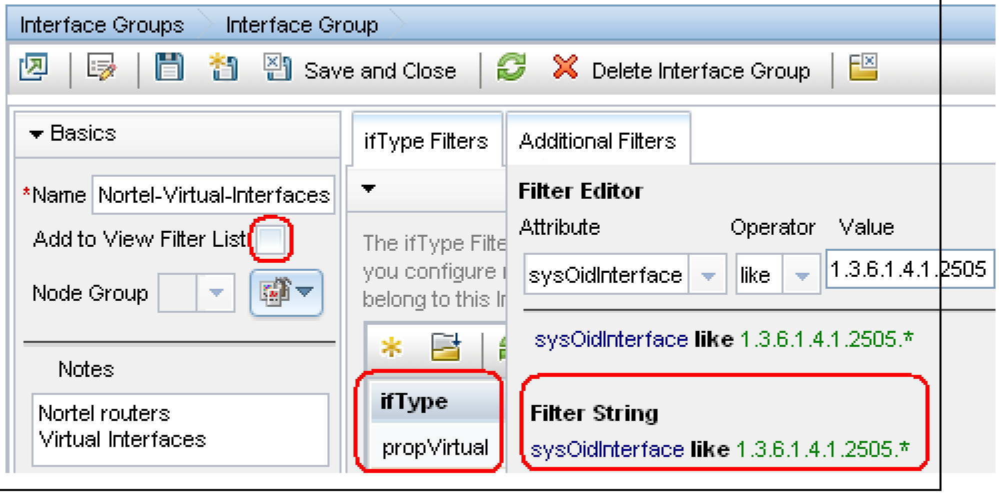
***Default Tenant* only**: If you have routers in your network domain that contain a large number of physical and virtual interfaces, you may want Auto-Discovery to only find and monitor the important interfaces.

Follow the instructions in "Set Outside Limits for Auto-Discovery" on page 228 and use the following choices in the appropriate steps:

Only Routers' Physical Interfaces Discovered

<table>
<tr><th>Task</th><th>How</th></tr>
<tr><td><b>Auto-Discovery Rule = Included</b></td><td><b>Discover Matching Nodes</b> ☑
Create one or more <b>IP Ranges</b> settings that identify the location of routers in your network domain (specify the area within Default Tenant that this Auto-Discovery Rule controls):
Enter <b>IP Range</b> < IPv4 / IPv6 range> (Minimum: One is required in one of your Auto-Discovery Rules.)
Set <b>Range Type</b> Include in rule ▾</td></tr>
<tr><td><b>Auto-Discovery Rule = Rejects</b></td><td></td></tr>
<tr><td>Spiral Discovery: For Routers that are in any Tenant:</td><td>If routers in your network have more than 2048 physical interfaces plus virtual interfaces, and you want Spiral Discovery to gather data about only a subset of those interfaces, create a <i>pair</i> of filters as follows:
• "Configure an Included Interface Ranges Filter" on page 251 (each entry based on one MIB-II <code>sysObjectID</code> and a range of <code>ifIndex</code> values)
• "Configure an Excluded Interfaces Filter" on page 254 (each entry based on a defined Interface Group)
For example, the following <i>pair</i> of filters could instruct Spiral Discovery to gather information about an <code>ifIndex</code> range representing the physical Nortel interfaces within that network environment, then reject any virtual interfaces that are an exception to that assumption:
• Included Interface Range defined as <code>sysObjectID = Nortel and ifIndex = 1-256</code>

<table><tr><th>System Object ID Prefix</th><th>Low ifIndex Value</th><th>High ifIndex Value</th><th>Notes</th></tr><tr><td>1.3.6.1.4.1.2505.</td><td>1</td><td>256</td><td>Nortel Routers</td></tr></table>
• Excluded Interfaces filter that instructs Spiral Discovery to ignore any exceptions to the assumption of the Included Interface Range defined in the Auto-Discovery Rule = Included configuration. Define an Interface Group that identifies Nortel</td></tr>
</table>

Only Routers' Physical Interfaces Discovered, continued

Task	How
	devices' Virtual Interfaces: **Tip:** The selected Interface Group will be empty after the next Spiral Discovery cycle. Consider disabling the Interface Group definition's **Add to View Filter List** □ attribute to prevent this empty Interface Group from appearing on selection lists within NNMi views.  Interface Groups / Interface Group Save and Close · Delete Interface Group Basics *Name: Nortel-Virtual-Interfaces Add to View Filter List □ Node Group Notes: Nortel routers Virtual Interfaces ifType Filters · Additional Filters The ifType Filte… you configure… belong to this I… ifType: propVirtual Filter Editor Attribute: sysOidInterface · Operator: like · Value: 1.3.6.1.4.1.2505 sysOidInterface **like** 1.3.6.1.4.1.2505.* **Filter String** sysOidInterface **like** 1.3.6.1.4.1.2505.*

For additional Auto-Discovery ideas:

Only Specific Vendor/Make/Models Discovered

***Default Tenant* only**: If you want Auto-Discovery to find only devices within Default Tenant that were manufactured by a specific vendor, you must use `SNMP sysObjectID` values. Navigate to the **Configuration** workspace, and select the **Device Profiles** view to see all known system object IDs at the time NNMi released. You can add a Device Profile if the one you need is not yet configured.

For example: Do not use dashes or asterisks (*) in your system object ID value. Do not use a period (.) as the first character. A partial entry becomes a wildcard:

- To include all HP devices, use the following prefix in configuration settings for the pair of Auto-Discovery Rules:
  `1.3.6.1.4.1.11` (prefix for all HP devices)
- To specify certain HP devices, use the appropriate numbers, such as:
  `1.3.6.1.4.1.11.2.3.7.1.10` = hpnetSwitch200
  `1.3.6.1.4.1.11.2.3.7.2.2` = hpicfRouterTR

Note: After you set your configuration according to these guidelines, when a new HP device is added to your network, you do not need to do anything. NNMi discovers it during the next discovery cycle if it matches the criteria you define.

Follow the instructions in "Set Outside Limits for Auto-Discovery" on page 228 and use the following choices in the appropriate steps:

Only Specific Vendor/Make/Models Discovered

Task	How
Auto-Discovery Rule = Included	• **Discover Matching Nodes** ☑ • **Discover Any SNMP Device** ☑ • **Discover Non-SNMP Devices** □ *Optional:* If you want to limit Auto-Discovery to only the vendor/make/models that you specify, do the easiest one of the following (for more information see "SNMP System Object ID Ranges for the Auto-Discovery Rule" on page 225): **Note:** Do not use dashes or asterisks (*) in your system object ID value. Do not use a period (.) as the first character. A partial entry becomes a wildcard. • Create one or more **System Object ID Ranges**. Your list *must include everything* you want Auto-Discovery to find. • Do nothing here but make changes to the **System Object ID Ranges** in the *Auto-Discovery Rule = Rejects* configuration. • Use a combination such as: *Auto-Discovery Rule = Included* configuration: Included = `1.3.6.1.4.1.11` (`HP`) *Auto-Discovery Rule = Rejects* configuration: `1.3.6.1.4.1.11.2.3.7.1.10` (`hpnetSwitch200`) `1.3.6.1.4.1.11.2.3.7.2.2` (`hpicfRouterTR`)
Auto-Discovery Rule = Rejects	*Optional:* Create one or more **System Object ID Ranges** that identify the vendor/make/models that you do not want Auto-Discovery to find. **Note:** Do not use dashes or asterisks (*) in your system object ID value. Do not use a period (.) as the first character. A partial entry becomes a wildcard. For example, `hpnetSwitch200` and `hpicfRouterTR`: **System Object ID Prefix** `1.3.6.1.4.1.11.2.3.7.1.10` **Range Type** Include in rule ▾ **System Object ID Prefix** `1.3.6.1.4.1.11.2.3.7.2.2` **Range Type** Include in rule ▾

For additional Auto-Discovery ideas:

All SNMP Devices Discovered

***Default Tenant* only**: If you want Auto-Discovery to automatically find all devices that respond to SNMP within Default Tenant, use these guidelines.

> **Note:** This strategy might cause you to reach your licensed capacity very quickly. See "Extend a Licensed Capacity" on page 1575.
>
> After you set your configuration according to these guidelines, when a new device is added to your network, you do not need to do anything. NNMi discovers it during the next discovery cycle if the device responds to SNMP queries.

Follow the instructions in "Set Outside Limits for Auto-Discovery" on page 228 and use the following choices in the appropriate steps:

All SNMP Devices Discovered

Task	How
Auto-Discovery Rule = Included	• **Discover Matching Nodes** ☑ • **Discover Any SNMP Device** ☑ • **Discover Non-SNMP Devices** □ Create one or more: **IP Range** < IPv4 / IPv6 range> (Minimum: One is required in one of your Auto-Discovery Rules.) **Range Type** Include in rule ▾ If you want Auto-Discovery to find all SNMP devices, do not create any **System Object ID Ranges**.
Auto-Discovery Rule = Rejects	

For additional Auto-Discovery ideas:

Everything Discovered

***Default Tenant* only**: If you want Auto-Discovery to automatically find all devices within Default Tenant, use these guidelines.

If the device does not support SNMP, NNMi queries DNS to determine the hostname. If this hostname matches another non-SNMP node, NNMi merges the information to create only one node with multiple associated addresses to preserve licensed capacity limits for discovered nodes. This is why the "Well-Configured DNS Prerequisite" on page 188 is very important.

> **Note:** This strategy might cause you to reach your licensed capacity very quickly. See

> "Extend a Licensed Capacity" on page 1575.
>
> After you set your configuration according to these guidelines, when a new device is added to your network, you do not need to do anything. NNMi discovers it during the next discovery cycle.

Follow the instructions in "Set Outside Limits for Auto-Discovery" on page 228 and use the following choices in the appropriate steps:

All SNMP Devices Discovered

Task	How
Auto-Discovery Rule = Included	• **Discover Matching Nodes** ☑ • **Discover Any SNMP Device** ☑ • **Discover Non-SNMP Devices** ☑ Create one or more: **IP Range** < IPv4 / IPv6 range> (Minimum: One is required in one of your Auto-Discovery Rules.) **Range Type** Include in rule ▾ If you want Auto-Discovery to find all SNMP devices, do not create any **System Object ID Ranges**.
Auto-Discovery Rule = Rejects	

For additional Auto-Discovery ideas:

Strategies to Exclude Certain Nodes from Auto-Discovery

***Default Tenant* only**: Sometimes it is useful to exclude certain nodes from Auto-Discovery and Monitoring. For example:

- All of your printers
- Certain problem devices

Techniques to exclude nodes include the following:

1. Follow the instructions in "Set Outside Limits for Auto-Discovery" on page 228 and in the appropriate steps, use any of the following choices required to clearly identify the Nodes that Auto-Discovery should exclude:

    a. Set up your **Auto-Discovery Rule = Included** IP Ranges without specifying any addresses from the problem nodes.

    b. Set up your **Auto-Discovery Rule = Rejects** settings to ignore information received about the problem nodes using either or both of the following:

> **Caution:** These settings instruct Auto-Discovery to not add the specified IP addresses or devices with a specified MIB-II `sysObjectID` to the NNMi database, not acknowledge any Hints received about them, nor gather Discovery Hints from them unless the address is a discovery seed. See "Specify Discovery Seeds" on page 256 to learn how to establish discovery seeds.
>
> If you want this behavior for Spiral Discovery in all Tenants, see "Configure an Excluded IP Addresses Filter" on page 248.

- Create any number of:

  IP Range
  < IPv4 / IPv6 range>

  Range Type
  Include in rule

- Create any number of:

  System Object ID Range
  < sysObjectID >

  Range Type
  Include in rule

  System Object ID Ranges enable you to identify the vendor/make/model of the devices that you do not want Auto-Discovery to find. For more information, see "Only Specific Vendor/Make/Models Discovered" on page 234.

For additional Auto-Discovery ideas:

2. If you want to prevent NNMi from generating *any* network traffic to certain Nodes, see "Configuring Communication Protocol" on page 119. Configure NNMi to never attempt any SNMP or ICMP communication with those Nodes.

For strategies to prevent specific devices from being discovered:

Limit Sources of Neighbor Information

***Default Tenant* only**: If you want Auto-Discovery to *never use* a particular IP address as a source for gathering additional information (using SNMP, ICMP, ARP cache, DNS, and a variety of other protocols), follow the instructions in "Set Outside Limits for Auto-Discovery" on page 228 and use the following choices in the appropriate steps:

Limit Auto-Discovery Hints

Task	How
Auto-Discovery Rule = Included	
Auto-Discovery Rule = Rejects	Create one or more **IP Ranges** settings that clearly identify the addresses. Auto-Discovery does not gather any Hints for further discovery from these addresses: Enter **IP Range** < IPv4 / IPv6 range>

Limit Auto-Discovery Hints, continued

Task	How
	Set **Range Type** Include in rule ▾ **Note:** Because the Auto-Discovery Rule = Reject's setting is **Discover Matching Nodes** □ disabled, Auto-Discovery *does not gather neighbor information* from the addresses identified in any IP address range included in this rule. The addresses, themselves, might still show up in the topology database because of the following: • Neighbor information is still gathered from IP addresses specifically identified in the discovery seeds configuration settings. • NNMi also uses the source IP address from SNMP traps as hints to discovery. NNMi uses those hint IP address only for initial discovery. NNMi then requests the current Management Address (the address from which the node's SNMP Agent responds) and uses that IP address for all communication. NNMi never gathers Auto-Discovery *Hints* from IP addresses assigned to a Tenant other than the Default Tenant. **Caution:** If you want to prevent NNMi from generating *any* network traffic to certain Nodes, see "Configuring Communication Protocol" on page 119.
Spiral Discovery: For devices in Tenants other than Default Tenant:	"Configure an Excluded IP Addresses Filter" on page 248 (based on IP address ranges) "Configure an Excluded Interfaces Filter" on page 254 (based on defined Interface Groups) "Configure an Included Interface Ranges Filter" on page 251 (based on one or more SNMP `sysObjectID` and `ifIndex` range values)

The IP addresses in the following table cannot be used as Discovery Seeds or Auto-Discovery Hints. NNMi still Discovers and Monitors these addresses within the context of a Node, but NNMi does not gather information about neighbors from these addresses.

Invalid IP Addresses for Discovery Seeds or Auto-Discovery Hints

IPv4 Address Range	IPv6 Address Range	Explanation
`0.*.*.*`	not applicable	Reserved IP addresses
`0.0.0.0`	`::0`	Any Local (listen) address
`127.*.*.*`	`::1`	Loopback addresses

Invalid IP Addresses for Discovery Seeds or Auto-Discovery Hints, continued

IPv4 Address Range	IPv6 Address Range	Explanation
not applicable	`fe80::*:*:*:*`	**IPv6 link-local address**[1]
`224-239.*.*.*`	not allowed (`ff00::` to `ffff:ffff:ffff:ffff:ffff:ffff:ffff`)	**multicast address**[2]
`255.255.255.255`	not applicable	Broadcast address

For additional Auto-Discovery ideas:

For strategies to prevent specific devices from being discovered:

Choose Techniques to Launch Discovery

Available choices for Auto-Discovery (within Default Tenant) and Spiral Discovery (all Tenants) are as follows:

Two techniques are available for launching Spiral Discovery:

- Provide a Discovery Seed to identify each Node you want NNMi to Discover.
- Auto-Discovery (in Default Tenant only): Configure either Discovery Seeds or Ping Sweep (ICMP ping), or both as starting points for Auto-Discovery. NNMi requests information about all known neighboring devices and then discovers the neighboring devices within the Default Tenant's address range.

  Ping Sweep works only with *IPv4 addresses*. In Wide Area Networks (WANs) such as ATM, Frame Relay, and Point-to-Point (where ARP cache is not available), the Ping Sweep locates nodes for Auto-Discovery to use when gathering neighbor information and evaluating "Consider IPv4 Subnet Connection Rules" on page 181.

  NNMi discovers any devices that comply with your Auto-Discovery Rule configurations and creates a record of each device in the NNMi database. If the device supports SNMP, all addresses for that device are combined into one Node object. If the device does not support

[1] A non-routable IPv6 unicast address only used for communication with other nodes on the same link (LAN or VLAN). Link local addresses cannot be used for communication that must be forwarded through a router. IPv6 auto-configuration automatically assigns a unique link local address in the fe80::/10 address space to each IPv6-enabled interface on a system.

[2] Used to identify a group of hosts joined into a group. IPv4 multicast addresses are in the range 224.0.0.0 to 239.255.255.255 and IPv6 multicast addresses have the prefix ff00::/8.

SNMP, NNMi queries DNS to determines the hostname. If this hostname matches another non-SNMP node, NNMi merges the information to create only one node with multiple associated addresses.

Two additional methods are possible for launching Discovery:

- NNMi administrators can initiate Discovery for a particular Node using the **Actions** → **Polling** → **Configuration Poll** menu item. See Using Actions to Perform Tasks for more information.

  Tip: You can right-click any object in a table or map view to access the **Actions** menu.

  Each time you select **Actions** → **Polling** → **Configuration Poll**, NNMi also applies any Custom Poller Policy to the nodes in its specified Node Group. This determines which instances should be polled. See "Create Custom Polling Configurations" on page 419 for more information.

- Auto-Discovery also uses the source IP address from SNMP traps as Discovery Hints for new addresses. If your Auto-Discovery Rules' IP Ranges include that new IP address, NNMi uses the Trap Hint for initial discovery of that address. NNMi then requests the Node's current Management Address (the address from which the node's SNMP Agent responds) and uses that IP address for all further communication. NNMi calculates whether the new address belongs to a previously discovered Node or a new Node.

Discovery Seeds for Auto-Discovery in Default Tenant

Discovery seeds are optional for the Nodes in the Default Tenant, but required for each Node assigned to any other Tenant.

Caution: If your network uses any of the following IPv4 translation protocols, you must create a unique Tenant (other than *Default Tenant*) for each domain of nodes with addresses determined by the following protocols (see "Overlapping Addresses in NAT Environments" on page 89):

- *Static* Network Address Translation (NAT)
- *Dynamic* Network Address Translation (NAT)
- *Dynamic* Port Address Translation (PAT/NAPT)

A discovery seed is a specific node that you want NNMi to discover. For example, a discovery seed might be a core router in your management environment.

Each discovery seed is identified by hostname (*not case-sensitive*) or IP address, and Initial Discovery Tenant assignment. When you add a discovery seed, NNMi immediately tries to discover that device (without waiting until the next regularly scheduled discovery interval). If discovery is not successful, NNMi tries again 10 minutes later, and continues trying. The time between each attempt is doubled until the time reaches 1 week or equals your current discovery interval.

NNMi discovers seed addresses regardless of how you configure Auto-Discovery Rule definitions or the Excluded IP Addresses filter.

Note: Nodes configured as discovery seeds are always discovered and added to the topology

> database. If you change your mind and delete a discovery seed configuration, the node is not automatically deleted from the topology database. See "Delete Nodes" on page 1602.

If you configure one or more Auto-Discovery Rules, note the following:

- If **Discover Matching Nodes** ☑ is enabled for an Auto-Discovery Rule, NNMi uses each discovery seed as a starting point to gather information about neighboring devices to expand discovery.

  > **Note:** You can use the Ping Sweep option in your Auto-Discovery Rules in addition to or instead of Discovery Seeds.

- If **Discover Matching Nodes** □ is disabled for an Auto-Discovery Rule, no devices matching that rule's criteria are discovered and added to the topology database unless:
  - The device's address is a discovery seed.

    See "Specify Discovery Seeds" on page 256 to learn how to establish discovery seeds.
  - The device's address is reported as a neighbor to another discovered address.

    If you want to ensure that an address is never added to the NNMi database, use the settings for "Configure an Excluded IP Addresses Filter" on page 248 or "Configure an Excluded Interfaces Filter" on page 254.

Ping Sweep for Auto-Discovery in Default Tenant

***Default Tenant* only**: You have two choices for Auto-Discovery starting points. Use either or both to best advantage for Nodes configured for the *Default Tenant* in your network environment:

- Discovery Seeds
  You designate specific hostnames (*not case-sensitive*) or IP addresses where Auto-Discovery starts gathering neighbor information.
- Ping Sweep
  NNMi issues ICMP pings to certain addresses gathered from neighbor information.

  > **Note:** Ping Sweep works only with IPv4 addresses. All nodes discovered using Auto-Discovery are assigned to the *Default Tenant*.

  Ping Sweep sends ICMP ping commands to IP addresses in the ranges defined in your Auto-Discovery rules. Ping Sweep enforces the following limits to the ICMP pings:

  - For each specific IP address range, NNMi issues pings across a maximum of the last two octets in the IPv4 address range. This is equivalent to a /16 subnet
  - ICMP pings are limited to 500 at one time. This avoids flooding your network or causing spam detection tools to set off an alarm.

Ping Sweep is useful in wide area networks such as ATM, Frame Relay, and Point-to-Point that do not contain an Address Resolution Protocol (ARP) cache.

You configure the Ping Sweep feature at two levels:

- "Configure Ping Sweep (override for all Auto-Discovery Rules)" on page 202
- "IP Address Ranges for the Auto-Discovery Rule" on page 221 (Ping Sweep configuration for each rule)

Spiral Discovery of Only Seeds (all Tenants)

Use these guidelines if any of the following are true:

- You want NNMi to discover only what you specify.
- Your network includes nodes with addresses provided by any of the following protocols (see "Overlapping Addresses in NAT Environments" on page 89):
  - *Static* Network Address Translation (NAT)
  - *Dynamic* Network Address Translation (NAT)
  - *Dynamic* Port Address Translation (PAT/NAPT)
- You want to control which Nodes each NNMi user sees. See "Tenant and Initial Discovery Security Group Assignments" on page 198.

> **Note:** After you set your configuration according to these guidelines, when a new device is added to your network, NNMi does not discover that device unless you configure another discovery seed to identify that device.

Configuration Steps to Discover Only What You Specify

Task	How
Do not include any **Auto-Discovery Rules**. **Note:** Auto-Discovery Rules can only be used to find devices assigned to the Default Tenant.	None are required for this strategy.
NNMi provides one *Default Tenant*. If you do not define any additional Tenants, all nodes belong to the Default Tenant and all NNMi users can see all Nodes within the Default Tenant. Configure a Tenant for each subset of devices you want to identify within your network environment for network segmentation or security purposes. NNMi users can then be assigned to the appropriate Tenant. See "Configuring Security" on page 503. **Caution:** If your network uses any of the following address translation protocols, you must create a unique Tenant (other than *Default Tenant*) for each domain of nodes with addresses determined by the following protocols (see "Overlapping Addresses in NAT Environments" on page 89): • *Static* Network Address Translation (NAT) • *Dynamic* Network Address Translation (NAT)	"Configure Tenants" on page 194.

Configuration Steps to Discover Only What You Specify, continued

Task	How
• *Dynamic* Port Address Translation (PAT/NAPT) Each member node must be identified with a discovery Seed configuration (see next row in this table). **Note:** All members of a Router Redundancy Group must be assigned to the same Tenant (visible in the Node form's Basic Attributes and in the Tenants column of the Inventory > Nodes view). The NNMi administrator configures the Tenants.	
In Discovery Configuration's **Seeds** view, for each device you want NNMi to discover: • Designate the hostname (*not case-sensitive*) or IP address. **Caution:** For nodes with addresses provided by Network Address Translation (NAT) protocols, use the appropriate address (see "Overlapping Addresses in NAT Environments" on page 89): ▪ *Static* Network Address Translation (NAT): ○ If the NNMi management server is outside the NAT domain - use the node's *external IP address* ○ If the NNMi management server is inside the NAT domain - use the node's *internal IP address* ▪ *Dynamic* Network Address Translation (NAT) - use the node's *internal IP address*. ▪ *Dynamic* Port Address Translation (PAT/NAPT) - use the node's *internal IP address*. For more information: • Designate the Tenant assignment if other than Default Tenant. Then configure NNMi to monitor your SNMP devices. See "Monitoring Network Health" on page 340.	"Specify Discovery Seeds" on page 256

Note: You control how often Spiral Discovery checks the discovered nodes based on a **Rediscovery Interval** setting. See "Adjust the Rediscovery Interval" on page 210 for more information.

Configure IPv4 Subnet Connection Rules

NNMi uses Subnet Connection Rules to detect connections between interfaces associated with IPv4 addresses that *do not respond* to Layer 2 *discovery protocols* (see the list of Topology Source protocols in Layer 2 Connection Form). Subnet Connection Rules take priority over the Layer 2 discovery protocol results. For special cases, you can override a Subnet Connection Rule by using the Connection Editor command line tool, see nnmconnedit.ovpl for more information.

NNMi provides a variety of predefined Subnet Connection Rules. For ideas, see "Subnet Connection Rules Provided by NNMi" on page 247.

Subnet Connection Rules are ideal for multiple situations. For additional details and examples of how Subnet Connection Rules work, see "Consider IPv4 Subnet Connection Rules" on page 181.

When Spiral Discovery detects a subnet, NNMi uses the matching Subnet Connection Rule to request information about all possible IPv4 addresses (potentially detecting previously undiscovered IPv4 addresses). NNMi checks the Excluded IP Addresses list. Any addresses in the list are dropped (for details, see "Configure an Excluded IP Addresses Filter" on page 248). Then NNMi creates connections among any interfaces associated with any newly discovered IPv4 addresses.

If important subnets in your network environment are not automatically connected by Spiral Discovery, edit a Subnet Connection Rule or create your own.

If you configure a Subnet Connection Rule, the rule independently applies to each Tenant. The members of Subnets must be unique Tenant/Node pairs (each Node assigned to only one Tenant). A Subnet Connection Rule can establish a link between the Default Tenant and another Tenant. However, links between two Tenants are not permitted *unless one of them is the Default Tenant*. See "Configure Tenants" on page 194.

To configure Subnet Connection Rules:

1. Complete all prerequisites. See "Prerequisites for Discovery" on page 187, .
2. Navigate to the **Subnet Connection Rule** form.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Expand **Discovery**.
   c. Select **Discovery Configuration**.
   d. Select the **Subnet Connection Rules** tab.
   e. Do one of the following:
      - To establish a rule, click the ✱ New icon, and continue.
      - To edit a rule, double-click the row representing the configuration you want to edit, and continue.
      - To delete a rule, select a row, and click the ✖ Delete icon.
3. Provide the required basic settings (see Basics table).
4. Provide the Subnet Connection behavior settings for this rule (see Details table).
5. Click **Save and Close** to return to the **Discovery Configuration** form.
6. Click **Save and Close** to apply the configuration. Spiral Discovery implements your changes during the next regularly scheduled discovery interval. If more than two nodes are connected using this rule, NNMi uses the following icon to indicate this special connection on maps (see example in "Consider IPv4 Subnet Connection Rules" on page 181):

If you double-click the icon, the Layer 2 Connection Form displays and the **Topology Source** value is `SUBNETCONNECTION`.

Basics for this Subnet Connection Rule

Task	How
Name	Type a meaningful name for this Subnet Connection Rule. Alpha-numeric and special characters (~ ! @ # $ % ^ & * ( ) _+ -) are permitted. No spaces are permitted. **Note:** This name is prepended to the Layer 2 connection name (when you request Tool Tips information about the connection on the Layer 2 Neighbor View map). If a subnet matches more than one rule, NNMi randomly chooses from among the matching rules.
Enable	If enabled ☑, NNMi uses the Subnet Connection Rule to create connections between interfaces associated with the IPv4 addresses within the specified subnets. If disabled □, NNMi ignores the Subnet Connection Rule.

Details for this Subnet Connection Rule

<table>
<tr><th>Task</th><th>How</th></tr>
<tr><td>Minimum IPv4 Prefix Length</td><td>Specify the minimum prefix length (subnet mask length) for the subnet where you want Spiral Discovery to create Layer 2 Connections. Spiral Discovery creates connections between interfaces associated with IPv4 addresses that have subnet prefix lengths equal to or greater than the specified value and meet the other specified criteria.
<table>
<tr><th>Valid Minimum IPv4 Prefix Length Values</th><th>Number of Usable IPv4 Addresses</th></tr>
<tr><td>28</td><td>14 (16-2=14)*</td></tr>
<tr><td>29</td><td>6 (8-2=6)*</td></tr>
<tr><td>30</td><td>2 (4-2=2)*</td></tr>
<tr><td>31</td><td>2</td></tr>
<tr><td colspan="2">* Two IPv4 addresses are reserved in each subnet. The first IPv4 address is used for the network itself and the last IPv4 address is reserved for broadcast.</td></tr>
</table>
</td></tr>
<tr><td>ifType</td><td><em>Optional</em>. Use this Interface MIB variable as an additional filter to identify the types of interfaces to include when creating the subnet connections. For example, if you want connections only between Frame Relay interfaces, select <code>frameRelay</code> as the ifType.</td></tr>
<tr><td>ifName</td><td><em>Optional</em>. Use this Interface MIB variable as an additional filter to identify the interfaces to include when creating the subnet connections. This attribute is useful if you have a naming convention that is used to identify a set of interfaces.</td></tr>
</table>

Details for this Subnet Connection Rule, continued

Task	How
	For example, `lan0`. Maximum 255 characters. The following wildcard characters are permitted: asterisk (*) represents any string, and question mark (?) represents a single character.
ifDescription	*Optional*. Use this Interface MIB variable as an additional filter to identify the interfaces to include when creating the subnet connections. For example, you might want to select a particular set of interfaces that have the same vendor description. Maximum 255 characters. The following wildcard characters are permitted: asterisk (*) represents any string, and question mark (?) represents a single character.
ifAlias	*Optional*. Use this Interface MIB variable as an additional filter to identify the interfaces to include when creating the subnet connections. This attribute is useful if you have an alias naming convention that is used to identify a set of interfaces. For example, `Connection to remote store in Hawaii`. Maximum 255 characters. The following wildcard characters are permitted: asterisk (*) represents any string, and question mark (?) represents a single character.

Subnet Connection Rules Provided by NNMi

The NNMi Subnet Connection Rules work only with IPv4 subnets.

NNMi provides the Subnet Connection Rules described in the following table (for more information, see "Consider IPv4 Subnet Connection Rules" on page 181).

The *Small Subnets* Rule ensures that NNMi detects IPv4 addresses within subnets of this size, regardless of the interface type. The remaining Subnet Connection Rules create connections based on interface type and the specified subnet size.

Tip: See "Consider IPv4 Subnet Connection Rules" on page 181 for more information about how Subnet Connection Rules use interface types.

To create new Subnet Connection Rules (or modify the ones provided), see "Configure IPv4 Subnet Connection Rules" on page 244.

If you configure a Subnet Connection Rule, the rule independently applies to each Tenant. The members of Subnets must be unique Tenant/Node pairs (each Node assigned to only one Tenant). A Subnet Connection Rule can establish a link between the Default Tenant and another Tenant. However, links between two Tenants are not permitted *unless one of them is the Default Tenant*. See "Configure Tenants" on page 194.

Subnet Connection Rules Provided by NNMi

Rule Name	Minimum IPv4 Prefix Length (Subnet Mask Length)	Interface Type (#)
Asynchronous Transfer Mode	28	atm (37)
Digital Signal 0	28	ds0 (81)
Digital Signal 1	28	ds1 (18)
Digital Signal 3	28	ds3 (30)
Digital Subscriber Loop over ISDN	28	idsl (154)
Frame Relay Interfaces	28	frameRelay (32)
Integrated Services Digital Network	28	isdn (63)
Multiprotocol Label Switching	28	mpls (166)
Point to Point	28	ppp (23)
Serial Line Internet Protocol	28	slip (28)
Serial Point to Point	28	propPointToPointSerial (22)
Small Subnets	30	
Synchronous Optical Networking	28	sonnet (39)

Configure an Excluded IP Addresses Filter

This configuration setting instructs NNMi to not add the specified IP addresses to the NNMi database (ignore that information when received from an SNMP agent), not acknowledge any Hints received about them, nor gather Discovery Hints from them, and delete them from the NNMi database during the next Spiral Discovery cycle (if previously discovered). Therefore, NNMi does not monitor or communicate with those addresses. See "Keep Requests to a Minimum" on page 183.

Caution: This filter applies to all nodes that meet the criteria within any Tenant.

Sometimes there are IP addresses or ranges of IP addresses in your environment that you do not want NNMi to discover or monitor. For example:

- There are multiple Nortel switches in your environment. They each have a non-routable IP address of 192.168.168.168 that is defined by the manufacturer. This special address is used to establish the default VLAN for the switch. However, NNMi discovers this duplicate address and establishes a lot of unnecessary connections on the Layer 3 Neighbor View map.

- Your service provider forbids the generation of ICMP or SNMP traffic from your NNMi installation. That range of addresses can easily be excluded to prevent violating your contractual agreement with the vendor.
- The Provider Edge (**PE**[1]) routers have addresses that NNMi ICMP ping commands cannot reach or have addresses that you want to exclude from Subnet views.

> **Note:** The node and interface associated with any address identified in your Excluded IP Address filter shows up in the topology database and maps. For information about excluding the entire node, see "Strategies to Exclude Certain Nodes from Auto-Discovery" on page 237.

Carefully select the addresses for your Excluded IP Addresses filter. Do not populate the Excluded IP Addresses filter with the addresses associated with SNMPv1/SNMPv2c agents or SNMPv3 engines (the Management Addresses).

> **Caution:** This filter applies to all nodes in all Tenants. If you exclude an IP address, any duplicates of that address in *static* Network Address Translation (NAT), *dynamic* Network Address Translation (NAT), or *dynamic* Port Address Translation (PAT/NAPT) domains of your network are also excluded. See "Overlapping Addresses in NAT Environments" on page 89.

> **Tip:** If you have a large number of IP addresses that you want to exclude from Spiral Discovery, see the nnmdiscocfg.ovpl Reference Page.

To exclude specific IP addresses from the discovery process:

1. Complete all prerequisites. See "Prerequisites for Discovery" on page 187, .
2. Navigate to the **Excluded IP Address** form.
   a. From the workspace navigation panel, select the 🔧 **Configuration** workspace.
   b. Expand **Discovery**.
   c. Select **Discovery Configuration**.
   d. Select the **Excluded IP Addresses** tab.
   e. Do one of the following:
      - To exclude an address or range of addresses from Spiral Discovery, click the ✱ New icon, and continue.
      - To edit an excluded address setting, click the 📂 Open icon in the row representing the configuration you want to edit, and continue.
      - To delete an excluded address setting, select a row, and click the ✖ Delete icon.
3. To specify a range of Excluded IP addresses, use one of the following. Pick one address notation style, combinations of wildcards and CIDR notation are not permitted within one

[1] Provider Edge router. The Internet Service Provider's router that receives your data on the path to your data's final desination. The Customer Edge (CE) router in your network connects to this PE.

address range. You can provide multiple address range settings:

- **IPv4 address wildcard notation**.

  An IPv4 Address range is a modified dotted-notation where each octet is one of the following:

  - A specific octet value between 0 and 255
  - A low-high range specification for the octet value (for example, "112-119")
  - An asterisk (*) wildcard character, which is equivalent to the range expression "0-255"

  > **Note:** The following two IPv4 addresses are considered invalid: `0.0.0.0` and `127.0.0.0`

  Examples of valid IPv4 address wildcards include:

  ```
  10.1.1.*
  10.*.*.*
  10.1.1.1-99
  10.10.50-55.*
  10.22.*.4 10.1-9.1-9.1-9
  ```

- **IPv4 Classless Inter-Domain Routing (CIDR) notation**.

  The CIDR notation specifies the number of consecutive bits in the IPv4 address that must match.

  For example, `10.2.120.0/21`

  > **Note:** NNMi does not support CIDR subnet mask notation such as, `10.2.120.0/255.255.248.0`

Example IPv4 Prefix Length Values	Number of Usable IPv4 Addresses
28	14 (16-2=14)*
29	6 (8-2=6)*
30	2 (4-2=2)*
31	2

  * Two IPv4 addresses are reserved in each subnet. The first IPv4 address is used for the network itself and the last IPv4 address is reserved for broadcast.

- **IPv6 address wildcard notation**

  Separate each 16-bit value of the IPv6 address with a colon. The 16-bit value can be any of the following:

  - A specific hexadecimal value between `0` and `FFFF` (case insensitive).
  - A low-high range specification of the hexadecimal value (for example, `1-1fe`).
  - An asterisk (`*`) wildcard character (equivalent to the range expression `0-ffff`).

> **Note:** The standard IPv6 short-hand notation (`::`) is allowed to express one or more 16-bit elements of zero (`0`) values. However, the mixed IPv6/IPv4 dot-notation (for example, `2001:d88::1.2.3.4`) is not permitted as an IPv6 address range.

Valid examples of ranges in modified IPv6 address notation include the following:

```
2001:D88:0:A00-AFF:*:*:*:*
2001:D88:1:*:*:*:*:*
2001:D88:2:0:a07:ffff:0a01:3200-37ff
```

- **IPv6 Classless Inter-Domain Routing (CIDR) notation**

  The CIDR notation specifies the number of consecutive bits in the IPv6 address that must match.

  `2001:d88:a00::/44` (equivalent to modified IPv6 address notation `2001:d88:a00-a0f:*:*:*:*:*`)

  For example, valid IPv6 address ranges in CIDR notation include the following:

  `2001:d88:0:a00::/56` (equivalent to modified IPv6 address notation `2001:D88:0:A00-AFF:*:*:*:*`)

  `2001:d88:1::/48` (equivalent to modified IPv6 address notation `2001:D88:1:*:*:*:*:*`)

4. Click **Save and Close**. Spiral Discovery implements your changes during the next regularly scheduled discovery interval. To apply the changes immediately, use **Actions** → **Polling** → **Configuration Poll**. See Using Actions to Perform Tasks for more information.

> **Tip:** You can right-click any object in a table or map view to access the **Actions** menu.

For strategies to prevent specific devices from being discovered:

Configure an Included Interface Ranges Filter

Sometimes there are certain types of interfaces in your environment that you want NNMi to discover. For example, you might have large devices with thousands of interfaces and want NNMi to discover and monitor only a subset of the interfaces in these devices.

This configuration setting instructs Spiral Discovery to only request data about a subset of Interfaces within the specified vendor/make/models (determined by MIBII `sysObjectID`) . See "Keep Requests to a Minimum" on page 183.

> **Caution:** This filter applies to all nodes that meet the criteria within any Tenant.

Rather than requiring that you specify each interface, NNMi enables you to use the System Object ID prefix (SNMP MIBII `sysObjectID`) and the `ifIndex` values to specify a range of interfaces that you want NNMi to discover. Use the **Configuration** → **Device Profiles** view to see the list of all known system object IDs at the time NNMi released.

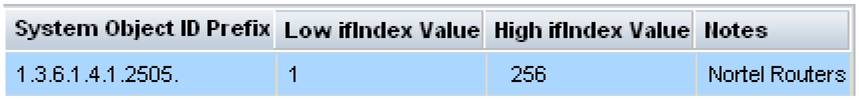
> **Tip:** To exclude any particular interfaces within that range you can also use the Excluded Interfaces tab. See"Configure an Excluded Interfaces Filter" on page 254 for more information.

To include Interfaces in the Spiral Discovery process using Included Interface Ranges:

1. Complete all prerequisites. See "Prerequisites for Discovery" on page 187, .
2. Navigate to the **Included Interface Ranges** tab.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Expand **Discovery**.
   c. Select **Discovery Configuration**.
   d. Select the **Included Interface Ranges** tab.
3. Do one of the following:
   - To specify an Interface Range to include in Spiral Discovery, click the New icon, and continue.
   - To edit an Included Interface Ranges setting, double-click the row representing the configuration you want to edit, and continue.
   - To delete an Included Interface Ranges setting, select a row, and click the Delete icon.
   - To refresh the list of Included Interface Ranges settings, click the Refresh icon.
4. Provide the required basic settings (see Basics table)

   Many routers have thousands of interfaces. If you want NNMi to actively discover and monitor a subset of those interfaces, consider an Included Interface Range that specifies only those interfaces you care about. For example:

   `sysObjectID = Nortel and ifIndex = 1-256` (or any range that reflects reality in your network environment)



System Object ID Prefix	Low ifIndex Value	High ifIndex Value	Notes
1.3.6.1.4.1.2505.	1	256	Nortel Routers

> **Caution:** Be careful about determining which Interfaces are important to your team. Make sure all key interfaces (such as Loopbacks) are in the specified `ifIndex` range.

5. Click **Save and Close**. Spiral Discovery implements your changes during the next regularly scheduled discovery interval. To apply the changes immediately, use **Actions** → **Polling** → **Configuration Poll**. See Using Actions to Perform Tasks for more information.

> **Tip:** You can right-click any object in a table or map view to access the **Actions** menu.

6. To further refine the Interfaces you want Spiral Discovery to discover, use the **Excluded Interfaces** tab.See "Configure an Excluded Interfaces Filter" on page 254 for more information.

NNMi first checks for any Included Interface Range filter and then ignores data about any interfaces that are specified using the Excluded Interfaces filter.

> **Tip:** You can configure multiple `ifIndex` ranges for the same System Object ID prefix.

Included Interface Ranges Basic Attributes

Attribute	Description
System Object ID Prefix	Enter a prefix of an SNMP system object ID, or enter the entire SNMP system object ID. NNMi finds the longest (or most specific) matching system Object ID value. This means you can define generic rules for certain device families and more specific rules for specific device types. **Note:** Do not use dashes or asterisks (*) in your system object ID value. Do not use a period (.) as the first character. A partial entry becomes a wildcard. For example, if you enter `1.3.6.1.4.1.11`, discovery finds all HP devices. If you enter `1.3.6.1.4.1.9`, discovery finds all Cisco devices. **Tip:** You can configure multiple ifIndex ranges for the same System Object ID Prefix. For example, to configure Included Interface Ranges filters that specify ifIndex values 10 through 20 and 40 through 50 for the same node, create two Included Interface Range configurations for the same System Object ID Prefix. In the first Included Interface Range Filter, use the Low ifIndex Value **10** and the High ifIndex Value **20**. Create a second Included Interface Range Filter using the Low ifIndex Value **40** and the High ifIndex Value **50**.
Low ifIndex Value	Enter the lowest ifIndex value for the range of Interfaces you want to include in Spiral Discovery. Note the following • The Low `ifIndex` Value must be equal to or greater than 1 and less than 2147483647. • This value must be less than the **High ifIndex Value**
High ifIndex Value	Enter the highest `ifIndex` value for the range of Interfaces you want to include in Spiral Discovery. Note the following • The High ifIndex Value must be greater than 1 and less than or equal to 2147483647. • This value must be greater than the **Low ifIndex Value**.

For strategies to prevent specific devices from being discovered:

Configure an Excluded Interfaces Filter

This configuration setting instructs NNMi to not add the specified Interfaces to the NNMi database (ignore that information when received from an SNMP agent), not acknowledge any Hints received about them, nor gather Discovery Hints from them. Therefore, NNMi does not monitor or communicate with those interfaces. See "Keep Requests to a Minimum" on page 183.

Caution: This filter applies to all nodes that meet the criteria within any Tenant.

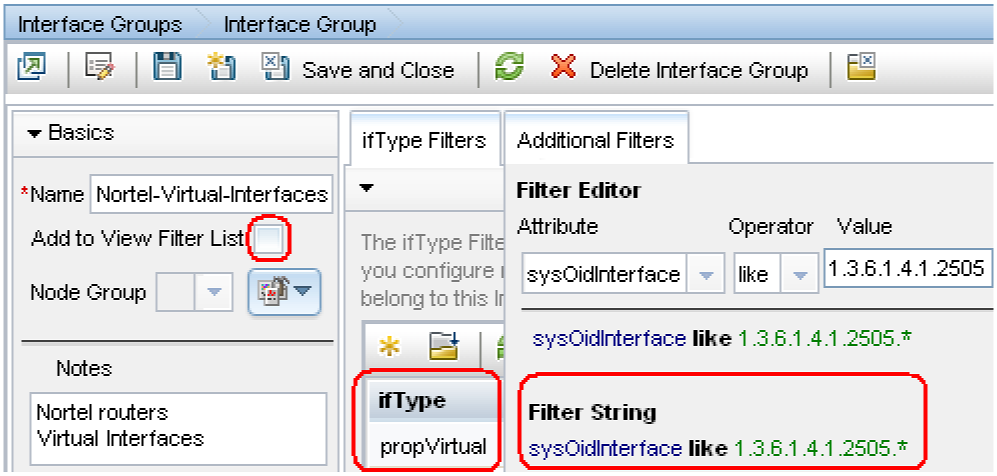
Once configured as an excluded interface:

- The interface's relationship to other objects is canceled:
  - Node
  - Address
  - VLAN Port
- The interface's membership status within any logical groups is removed:
  - Layer 2 Connections with **Link Aggregation**[1] (*NNMi Advanced*)
  - Router Redundancy Groups (*NNMi Advanced*)
  - VLANs
- During the next discovery cycle, NNMi automatically removes any previously discovered data associated with an excluded interface.

Note: The node and addresses associated with any interface identified in your Excluded Interface filter still shows up in the topology database and maps. For information about excluding the entire node, see "Strategies to Exclude Certain Nodes from Auto-Discovery" on page 237.

An Interface Group definition sets the criteria for exclusion. You can define Interface Groups using a wide range criteria choices. See "Create Interface Groups" on page 321. For example, the following Interface Group when used in an Excluded Interfaces filter instructs Spiral Discovery to ignore any Nortel routers' Virtual Interfaces. The selected Interface Group will be empty after the next Spiral Discovery cycle. Consider disabling the Interface Group definition's **Add to View Filter List** □ attribute to prevent this empty Interface Group from appearing on selection lists within NNMi views:

[1] Protocols used on Switches to configure multiple Interfaces (Aggregation Member Interfaces) to function as if they were one (an Aggregator Interface). When two Aggregator Interfaces establish a connection, that connection is an Aggregator Layer 2 Connection. The Aggregator Layer 2 Connection appears on Layer 2 Neighbor View maps as a thick line with an Interface icon at each end (representing the Aggregator Interface). When you double-click the thick line, it converts into multiple thin lines representing the participating Aggregation Member Layer 2 Connections with their Aggregation Member Interfaces at each end of the lines.



*Name: Nortel-Virtual-Interfaces

Notes: Nortel routers
Virtual Interfaces

ifType: propVirtual

Attribute: sysOidInterface Operator: like Value: 1.3.6.1.4.1.2505

sysOidInterface **like** 1.3.6.1.4.1.2505.*

Filter String
sysOidInterface **like** 1.3.6.1.4.1.2505.*

Be careful to not exclude Interfaces that are important to your team.

Note: Your Excluded Interfaces filter can be used in combination with an Included Interface Ranges filter. This strategy keeps network traffic to a minimum. The Included Interface Ranges use RFC 1213, MIB-II `sysObjectID` values paired with `ifIndex` ranges. Spiral Discovery then *requests only information about that subset of Interfaces from a matching Node's SNMP agent* (see "Configure an Included Interface Ranges Filter" on page 251).

If your Nodes have a high interface count and you want NNMi to Discover and Monitor only a subset of the most important Interfaces, consider using the Included Interface Ranges settings to identify the subset of important interfaces. Then your Excluded Interfaces Filter can instruct Spiral Discovery to reject a few items from within the included `ifIndex` ranges.

To exclude specific types of interfaces during the Spiral Discovery process:

1. Complete all prerequisites. See "Prerequisites for Discovery" on page 187, .
2. Navigate to the **Excluded Interfaces** tab.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Expand **Discovery**.
   c. Select **Discovery Configuration**.
   d. Select the **Excluded Interfaces** tab.
3. Do one of the following:
   - To select an Interface Group to filter certain interfaces out of Spiral Discovery, click the New icon, and continue.
   - To edit an excluded interfaces setting, double-click the row representing the configuration you want to edit, and continue.
   - To delete an excluded interfaces setting, select a row, and click the Delete icon.
   - To refresh the list of excluded interface settings, click the Refresh icon.
4. In the Interface Filter form, click the Lookup icon and select one of the options from the

drop-down menu:

- Show Analysis to view Analysis Pane information for the currently selected Interface Group. (See Use the Analysis Pane for more information about the Analysis Pane.)
- Quick Find to view and select from the list of all existing Interface Groups (for more information see "Use the Quick Find Window" on page 41).
- Open to display the details of the currently selected Interface Group.
- New to create a new Interface Group (see "Create Interface Groups" on page 321 for more information).

5. Click **Save and Close**. Spiral Discovery implements your changes during the next regularly scheduled discovery interval. To apply the changes immediately, use **Actions** → **Polling** → **Configuration Poll**. See Using Actions to Perform Tasks for more information.

> **Tip:** You can right-click any object in a table or map view to access the **Actions** menu.

For strategies to prevent specific devices from being discovered:

Specify Discovery Seeds

To configure discovery seeds do one or more of the following:

A discovery seed is a specific node that you want NNMi to discover.

Discovery seeds are sometimes optional and sometimes required. Before you begin, review the following topics:

- "Which Nodes Are Discovered?" on page 176
- "Configure Auto-Discovery Rules" on page 215
- "Determine Your Security Strategy" on page 507

Nodes specified as discovery seeds are always discovered and added to the topology database. As soon as you enter one or more discovery seeds, discovery begins. As part of the seed configuration, you specify a Tenant attribute value (and indirectly a Security Group attribute value). See "Configure Tenants" on page 194 for more information.

***Default Tenant* only**: If you create Auto-Discovery Rules, NNMi automatically gathers Hints from each discovered Node and uses that information to find any neighboring devices within your Default Tenant's address range.

If you want to use Auto-Discovery within the Default Tenant:

- Configure at least one Auto-Discovery Rule. See "Configure Auto-Discovery Rules" on page 215.
- Configure any number of Auto-Discovery Rules to maintain fine control over the scope of Auto-Discovery within the Default Tenant.

A discovery seed is a hostname (*not case-sensitive*) or IP address. Consider devices with the largest neighbor data in your network environment. For example, a good choice for a discovery seed would be a core router connected to a network you want to discover.

If you change your mind and delete a discovery seed from Discovery Configuration, the corresponding node is not deleted from the topology database. See "Delete Nodes" on page 1602 for information about removing the entire node record from the topology database.

Within the Default Tenant, Auto-Discovery can also use Ping Sweep instead of or in addition to discovery seeds to gather this neighbor information. See "Ping Sweep for Auto-Discovery in Default Tenant" on page 242 and "Discovery Seeds for Auto-Discovery in Default Tenant" on page 241.

> **Note:** Ping Sweep works only with IPv4 addresses. All nodes discovered using Auto-Discovery are assigned to the *Default Tenant*.

Related Topics

"Discovery Seed Results" on page 270

"Delete Discovery Seeds" on page 280

In the Console, Configure Discovery Seeds

Discovery seeds are sometimes optional and sometimes required. See "Specify Discovery Seeds" on the previous page for details.

Other methods of creating Discovery Seeds are "With a Seed File, Add Multiple Discovery Seeds" on page 262 and "From the Command Line, Add Discovery Seeds" on page 265.

To add a discovery seed using the console:

1. Complete all prerequisites. See "Prerequisites for Discovery" on page 187, .
2. Navigate to the **Seeds** view.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Expand **Discovery**.
   c. Select **Seeds**.
3. Do one of the following:
   - To add a discovery seed, click the ✳ New icon.
   - To edit a discovery seed, double-click the row representing the discovery seed you want to edit.

- To delete a discovery seed, select a row, and click the ✖ Delete icon (see "Delete Discovery Seeds" on page 280 and "Delete Nodes" on page 1602 for more information).

4. Provide appropriate information (see table).

   NNMi uses information gathered from Routers to establish membership for Subnet connections. Make sure that important Routers in your network environment are SNMP enabled.

   NNMi uses either of the following criteria to identify a Router:

   - The Router responds to an SNMP query with appropriate values for `sysServices` (1.3.6.1.2.1.1.7) and `ipForwarding` (1.3.6.1.2.1.4.1). See RFC 1213, MIB-II for details.
   - The Router responds to an SNMP query with an appropriate MIB-II `sysObjectID` value according to the current settings in NNMi's Device Profile configuration.

   You must provide the appropriate SNMP Community Strings to NNMi. See "Configuring Communication Protocol" on page 119.

5. Click **Save and Close** to return to the Discovery Configuration form.

   > **Tip:** Click the Save and New icon to continue to adding discovery seeds.

6. Click **Save and Close**. As soon as you enter one or more discovery seeds, discovery begins.

Discovery Seed Definition

Attribute	Definition
Hostname / IP Address	To identify the node, enter one of the following: • **Fully-qualified hostname** of the discovery seed (*not case-sensitive*) • **IP address** of the discovery seed If you specify an IP address, NNMi uses that IP address only during initial discovery of the Seed. NNMi then requests the current Management Address (the address from which the node's SNMP Agent responds) and uses that IP address for all communication after initial discovery. **Caution:** For nodes with addresses provided by Network Address Translation (NAT) protocols, use the appropriate address (see "Overlapping Addresses in NAT Environments" on page 89): • *Static* Network Address Translation (NAT): ▪ If the NNMi management server is outside the NAT domain - use the node's *external IP address* ▪ If the NNMi management server is inside the NAT domain - use the node's *internal IP address*

Discovery Seed Definition , continued

<table>
<tr><th>Attribute</th><th>Definition</th></tr>
<tr><td></td><td>• <i>Dynamic</i> Network Address Translation (NAT) - use the node's <i>internal IP address</i>.
• <i>Dynamic</i> Port Address Translation (PAT/NAPT) - use the node's <i>internal IP address</i>.
For more information:
When providing IPv6 addresses as discovery seeds, use IPv6 notation as defined in RFC 2373. Click here for more information.
• 16-byte (128-bit) address, composed of eight groups of 2-byte (16-bit) hex values separated by colons (XXXX:XXXX: XXXX:XXXX: XXXX:XXXX: XXXX:XXXX)
• Uppercase and lowercase (A-F/a-f) permitted for the hex digits.
<b>Note:</b> NNMi displays IPv6 addresses as all lowercase.
• <i>Optional</i>. Omit leading zeros in each 2-byte hex value.
• <code>::</code> means a single contiguous sequence of all zero 2-byte hex values. However, <code>::</code> is permitted only one time per address. For example, the following three IPv6 address notations are equivalent:
<code>2001:0D88:0000:0000:0008:0800:200C:417A</code>
<code>2001:d88:0:0:8:800:200c:417a</code>
<code>2001:d88::8:800:200C:417a</code>
• For the right-most 32-bits, IPv4 dotted-decimal notation can replace the pair of 2-byte hex values. For example, the following two IPv6 address notations are equivalent:
<code>2001:D88::5efe:10.7.150.201</code>
<code>2001:D88::5efe:a07:96c9</code>
<b>Types of IPv6 Addresses</b>
<table>
<tr><th>IPv6 Address Range</th><th>Explanation</th></tr>
<tr><td><code>0::</code>to <code>1fff:ffff:ffff:ffff:ffff:ffff:ffff</code></td><td>unassigned or reserved</td></tr>
</table>
</td></tr>
</table>

Discovery Seed Definition , continued

Attribute	Definition

Types of IPv6 Addresses, continued

IPv6 Address Range	Explanation
`2000::` to `3fff:ffff:ffff:ffff:ffff:ffff:ffff`	global unicast address[1]
`fd00::` to `fdff:ffff:ffff:ffff:ffff:ffff:ffff`	unique local address[2]

The IP addresses in the following table cannot be used as Discovery Seeds or Auto-Discovery Hints. NNMi still Discovers and Monitors these addresses within the context of a Node, but NNMi does not gather information about neighbors from these addresses.

Invalid IP Addresses for Discovery Seeds or Auto-Discovery Hints

IPv4 Address Range	IPv6 Address Range	Explanation
`0.*.*.*`	not applicable	Reserved IP addresses
`0.0.0.0`	`::0`	Any Local (listen) address
`127.*.*.*`	`::1`	Loopback addresses
not applicable	`fe80::*:*:*:*`	IPv6 link-

[1] (2000:: to 3fff:ffff:ffff:ffff:ffff:ffff:ffff:ffff) A publically routable IPv6 unicast address, used for communication between nodes anywhere on the internet. The first part of the address is a global routing prefix in the 2000::/3 address space for your organization (assigned by the Internet Service Providers). The complete host address can either be manually configured or automatically assigned using IPv6 auto-configuration and neighbor discovery.

[2] (fd00:: to fdff:ffff:ffff:ffff:ffff:ffff:ffff:ffff) A privately routable IPv6 unicast address used only for communication between nodes within your organization. The unique local addresses cannot be routed to the public internet. The address consists of a routing prefix in the fd00:/8 address spaces, assigned locally by your organization. And the full host address might be manually configured or automatically assigned using IPv6 auto-configuration and neighbor discovery.

Discovery Seed Definition , continued

<table>
<tr><th>Attribute</th><th>Definition</th></tr>
<tr><td></td><td>

Invalid IP Addresses for Discovery Seeds or Auto-Discovery Hints, continued

<table>
<tr><th>IPv4 Address Range</th><th>IPv6 Address Range</th><th>Explanation</th></tr>
<tr><td></td><td></td><td>local address[1]</td></tr>
<tr><td><code>224-239.*.*.*</code></td><td>not allowed (<code>ff00::</code> to <code>ffff:ffff:ffff:ffff:ffff:ffff:ffff</code>)</td><td>multicast address[2]</td></tr>
<tr><td><code>255.255.255.255</code></td><td>not applicable</td><td>Broadcast address</td></tr>
</table>

</td></tr>
<tr><td>Initial Discovery Tenant</td><td>

Optional. By default, NNMi assigns each Node to the *Default Tenant* and *Default Security Group*. See "Configure Tenants" on page 194 and "About Security Groups" on page 515 for more information. If you do not specify a Tenant, NNMi assigns this seed to the *Default Tenant* (and whichever Initial Discovery Security Group attribute value is currently configured for the Default Tenant).

Use the Initial Discovery Tenant setting to specify a Tenant for a particular seed, before discovery.

- To change the Initial Discovery Tenant, begin to type a valid Tenant name or Tenant **UUID**[3] and use the auto-complete feature to select the Tenant.

> **Tip:** You can also click the Lookup icon and select Quick Find from the Lookup field drop-down list. This option is useful when you want to see more than the Tenant name when determining which Tenant to select.

- To create a new Tenant, in the Lookup field, select New.

</td></tr>
<tr><td>Discovery Seed Results</td><td>An automatically generated value. The most recent discovery status for this discovery seed. See "Discovery Seed Results" on page 270 for details.</td></tr>
<tr><td>Last Modified</td><td>The date and time of the last change in Discovery Seed Results.</td></tr>
</table>

[1] A non-routable IPv6 unicast address only used for communication with other nodes on the same link (LAN or VLAN). Link local addresses cannot be used for communication that must be forwarded through a router. IPv6 auto-configuration automatically assigns a unique link local address in the fe80::/10 address space to each IPv6-enabled interface on a system.

[2] Used to identify a group of hosts joined into a group. IPv4 multicast addresses are in the range 224.0.0.0 to 239.255.255.255 and IPv6 multicast addresses have the prefix ff00::/8.

[3] Universally Unique Object Identifier, which is unique across all databases.

Discovery Seed Definition , continued

Attribute	Definition
Notes	Provide any additional information about this discovery seed that would be useful to you or your team. Type a maximum of 1024 characters. Alpha-numeric, spaces, and special characters (~ ! @ # $ % ^ & * ( ) _+ -) are permitted.

With a Seed File, Add Multiple Discovery Seeds

Discovery seeds are sometimes optional and sometimes required. See "Specify Discovery Seeds" on page 256 for details.

Other methods of creating Discovery Seeds are "In the Console, Configure Discovery Seeds " on page 257 and "From the Command Line, Add Discovery Seeds" on page 265.

Use a seed file to simultaneously add large numbers of discovery seeds. Your seed file contains one line for each discovery seed and, optionally, the Tenant to which the node belongs. If you do not specify a Tenant, NNMi assigns the node to the **Default Tenant**. See "Configuring Security" on page 503 and "Configure Tenants" on page 194 for more information.

For example:

```
12.2.111.104# cisco5500, "Hewlett_Packard"
12.2.112.268# cisco6509
12.2.119.205# cisco5500, "Hewlett_Packard"
```

Note: Any comments included after the # in a seed file become Notes attribute values for the discovery seeds.

To identify a discovery seed, enter one of the following:

- **Fully-qualified hostname** of the discovery seed (*not case-sensitive*)
- **IP address** of the discovery seed

  If you specify an IP address, NNMi uses that IP address only during initial discovery of the Seed. NNMi then requests the current Management Address (the address from which the node's SNMP Agent responds) and uses that IP address for all communication after initial discovery.

> **Caution:** For nodes with addresses provided by Network Address Translation (NAT) protocols, use the appropriate address (see "Overlapping Addresses in NAT Environments" on page 89):
>
> - *Static* Network Address Translation (NAT):
>   - If the NNMi management server is outside the NAT domain - use the node's *external IP address*
>   - If the NNMi management server is inside the NAT domain - use the node's *internal IP address*
> - *Dynamic* Network Address Translation (NAT) - use the node's *internal IP address*.

> - *Dynamic* Port Address Translation (PAT/NAPT) - use the node's *internal IP address*.
>
> For more information:

When providing IPv6 addresses as discovery seeds, use IPv6 notation as defined in RFC 2373. Click here for more information.

- 16-byte (128-bit) address, composed of eight groups of 2-byte (16-bit) hex values separated by colons (XXXX:XXXX: XXXX:XXXX: XXXX:XXXX: XXXX:XXXX)
- Uppercase and lowercase (A-F/a-f) permitted for the hex digits.

> **Note:** NNMi displays IPv6 addresses as all lowercase.

- *Optional*. Omit leading zeros in each 2-byte hex value.
- `::` means a single contiguous sequence of all zero 2-byte hex values. However, `::` is permitted only one time per address. For example, the following three IPv6 address notations are equivalent:

```
2001:0D88:0000:0000:0008:0800:200C:417A
2001:d88:0:0:8:800:200c:417a
2001:d88::8:800:200C:417a
```

- For the right-most 32-bits, IPv4 dotted-decimal notation can replace the pair of 2-byte hex values. For example, the following two IPv6 address notations are equivalent:

```
2001:D88::5efe:10.7.150.201
2001:D88::5efe:a07:96c9
```

Types of IPv6 Addresses

IPv6 Address Range	Explanation
`0::`to `1fff:ffff:ffff:ffff:ffff:ffff:ffff`	unassigned or reserved
`2000::` to `3fff:ffff:ffff:ffff:ffff:ffff:ffff`	**global unicast address**[1]
`fd00::` to `fdff:ffff:ffff:ffff:ffff:ffff:ffff`	**unique local address**[2]

The IP addresses in the following table cannot be used as Discovery Seeds or Auto-Discovery Hints. NNMi still Discovers and Monitors these addresses within the context of a Node, but NNMi does not gather information about neighbors from these addresses.

[1] (2000:: to 3fff:ffff:ffff:ffff:ffff:ffff:ffff:ffff) A publically routable IPv6 unicast address, used for communication between nodes anywhere on the internet. The first part of the address is a global routing prefix in the 2000::/3 address space for your organization (assigned by the Internet Service Providers). The complete host address can either be manually configured or automatically assigned using IPv6 auto-configuration and neighbor discovery.

[2] (fd00:: to fdff:ffff:ffff:ffff:ffff:ffff:ffff:ffff) A privately routable IPv6 unicast address used only for communication between nodes within your organization. The unique local addresses cannot be routed to the public internet. The address consists of a routing prefix in the fd00:/8 address spaces, assigned locally by your organization. And the full host address might be manually configured or automatically assigned using IPv6 auto-configuration and neighbor discovery.

Invalid IP Addresses for Discovery Seeds or Auto-Discovery Hints

IPv4 Address Range	IPv6 Address Range	Explanation
`0.*.*.*`	not applicable	Reserved IP addresses
`0.0.0.0`	`::0`	Any Local (listen) address
`127.*.*.*`	`::1`	Loopback addresses
not applicable	`fe80::*:*:*:*`	**IPv6 link-local address**[1]
`224-239.*.*.*`	not allowed (`ff00::` to `ffff:ffff:ffff:ffff:ffff:ffff:ffff`)	**multicast address**[2]
`255.255.255.255`	not applicable	Broadcast address

NNMi uses information gathered from Routers to establish membership for Subnet connections. Make sure that important Routers in your network environment are SNMP enabled.

NNMi uses either of the following criteria to identify a Router:

- The Router responds to an SNMP query with appropriate values for `sysServices` (1.3.6.1.2.1.1.7) and `ipForwarding` (1.3.6.1.2.1.4.1). See RFC 1213, MIB-II for details.
- The Router responds to an SNMP query with an appropriate MIB-II `sysObjectID` value according to the current settings in NNMi's Device Profile configuration.

You must provide the appropriate SNMP Community Strings to NNMi. See "Configuring Communication Protocol" on page 119.

To create a seed file:

In a text editor, type each entry on a separate line in the following format:

`<`*`IP_address`*`>` or `<`*`hostname`*`>, "<`*`tenant`*`>" #(optional comment to help identify the node)`

- <*IP_address*> = the IP address of the node
- <*hostname*> = the DNS fully-qualified or short hostname (*not case-sensitive*) of the node
- "<*tenant*>" = *Optional*. The name or **UUID**[3] of the Tenant to which the seeds are assigned. If

[1] A non-routable IPv6 unicast address only used for communication with other nodes on the same link (LAN or VLAN). Link local addresses cannot be used for communication that must be forwarded through a router. IPv6 auto-configuration automatically assigns a unique link local address in the fe80::/10 address space to each IPv6-enabled interface on a system.

[2] Used to identify a group of hosts joined into a group. IPv4 multicast addresses are in the range 224.0.0.0 to 239.255.255.255 and IPv6 multicast addresses have the prefix ff00::/8.

[3] Universally Unique Object Identifier, which is unique across all databases.

you do not provide a Tenant, NNMi assigns each node in the seed file to the Default Tenant. See "Configure Tenants" on page 194 for more information.

Tip: If you have two or more Tenants with the same name, use the UUID to specify the Tenant. To determine the UUID for a selected Tenant, open the Tenant form.

To add discovery seeds by loading a seed file:

Use the `nnmloadseeds.ovpl` command:

<path>/<file_name> = the name of the file that contains your discovery seeds

Windows:
```
%NnmInstallDir%\bin\nnmloadseeds.ovpl -f <path>\<file_name>
```

To assign all seeds within the seed file to one Tenant:
```
%NnmInstallDir%\bin\nnmloadseeds.ovpl -f <path>\<file_name> -t
<tenant_name>
```

UNIX:
```
/opt/OV/bin/nnmloadseeds.ovpl -f <path>/<file_name>
```

To assign all seeds within the seed file to one Tenant:
```
/opt/OV/bin/nnmloadseeds.ovpl -f <path>/<file_name> -t <tenant_name>
```

A message displays, showing the number of added, invalid, and ignored discovery seeds. For example:

```
26 seeds added
0 seeds invalid
0 seeds duplicated
```

See the nnmloadseeds.ovpl Reference Page for more information.

Related Topics

"Discovery Seed Results" on page 270

"Delete Discovery Seeds" on page 280

From the Command Line, Add Discovery Seeds

Discovery seeds are sometimes optional and sometimes required. See "Specify Discovery Seeds" on page 256 for details.

Other methods of creating Discovery Seeds are "In the Console, Configure Discovery Seeds " on page 257 and "With a Seed File, Add Multiple Discovery Seeds" on page 262.

You can add optional discovery seeds using the nnmloadseeds.ovpl command:

`<seed_list>` = the discovery seed entries (fully-qualified DNS hostname, short DNS hostname, or IP address)

Note: You can also specify the Tenant assignment for each discovery seed. If you do not specify a Tenant, NNMi assigns the node to the **Default Tenant**. See "Configuring Security" on page 503, "Configure Tenants" on page 194 and nnmloadseeds.ovpl for more information.

Windows:
```
%NnmInstallDir%\bin\nnmloadseeds.ovpl -n <seed_list>
```

To assign all seeds to one Tenant:
`%NnmInstallDir%\bin\nnmloadseeds.ovpl -n <seed_list> -t <tenant_name>`

UNIX:
`/opt/OV/bin/nnmloadseeds.ovpl -n <seed_list>`

To assign all seeds to one Tenant:
`/opt/OV/bin/nnmloadseeds.ovpl -n <seed_list> -t <tenant_name>`

In the following example, the devices with a hostname of cisco4 and cisco5, and a device with the IP address of 12.6.91.5 are added as discovery seeds and assigned to the Tenant named Hewlett_ Packard.

```
nnmloadseeds.ovpl -n cisco4 cisco5 12.6.91.5 -t Hewlett_Packard
```

Note: Identify the discovery seed by either a DNS-resolvable hostname or an IP address.

When adding individual discovery seeds using the **nnmloadseeds.ovpl** command:

- **Fully-qualified hostname** of the discovery seed (*not case-sensitive*)
- **IP address** of the discovery seed

  If you specify an IP address, NNMi uses that IP address only during initial discovery of the Seed. NNMi then requests the current Management Address (the address from which the node's SNMP Agent responds) and uses that IP address for all communication after initial discovery.

  > **Caution:** For nodes with addresses provided by Network Address Translation (NAT) protocols, use the appropriate address (see "Overlapping Addresses in NAT Environments" on page 89):
  >
  > - *Static* Network Address Translation (NAT):
  >   - If the NNMi management server is outside the NAT domain - use the node's *external IP address*
  >   - If the NNMi management server is inside the NAT domain - use the node's *internal IP address*
  > - *Dynamic* Network Address Translation (NAT) - use the node's *internal IP address*.
  > - *Dynamic* Port Address Translation (PAT/NAPT) - use the node's *internal IP address*.
  >
  > For more information:

When providing IPv6 addresses as discovery seeds, use IPv6 notation as defined in RFC 2373. Click here for more information.

- 16-byte (128-bit) address, composed of eight groups of 2-byte (16-bit) hex values separated by colons (XXXX:XXXX: XXXX:XXXX: XXXX:XXXX: XXXX:XXXX)
- Uppercase and lowercase (A-F/a-f) permitted for the hex digits.

  > **Note:** NNMi displays IPv6 addresses as all lowercase.

- *Optional*. Omit leading zeros in each 2-byte hex value.

- `::` means a single contiguous sequence of all zero 2-byte hex values. However, `::` is permitted only one time per address. For example, the following three IPv6 address notations are equivalent:
  `2001:0D88:0000:0000:0008:0800:200C:417A`
  `2001:d88:0:0:8:800:200c:417a`
  `2001:d88::8:800:200C:417a`

- For the right-most 32-bits, IPv4 dotted-decimal notation can replace the pair of 2-byte hex values. For example, the following two IPv6 address notations are equivalent:
  `2001:D88::5efe:10.7.150.201`
  `2001:D88::5efe:a07:96c9`

Types of IPv6 Addresses

IPv6 Address Range	Explanation
`0::`to `1fff:ffff:ffff:ffff:ffff:ffff:ffff`	unassigned or reserved
`2000::` to `3fff:ffff:ffff:ffff:ffff:ffff:ffff`	**global unicast address**[1]
`fd00::` to `fdff:ffff:ffff:ffff:ffff:ffff:ffff`	**unique local address**[2]

The IP addresses in the following table cannot be used as Discovery Seeds or Auto-Discovery Hints. NNMi still Discovers and Monitors these addresses within the context of a Node, but NNMi does not gather information about neighbors from these addresses.

Invalid IP Addresses for Discovery Seeds or Auto-Discovery Hints

IPv4 Address Range	IPv6 Address Range	Explanation
`0.*.*.*`	not applicable	Reserved IP addresses
`0.0.0.0`	`::0`	Any Local (listen) address
`127.*.*.*`	`::1`	Loopback addresses

[1] (2000:: to 3fff:ffff:ffff:ffff:ffff:ffff:ffff:ffff) A publically routable IPv6 unicast address, used for communication between nodes anywhere on the internet. The first part of the address is a global routing prefix in the 2000::/3 address space for your organization (assigned by the Internet Service Providers). The complete host address can either be manually configured or automatically assigned using IPv6 auto-configuration and neighbor discovery.

[2] (fd00:: to fdff:ffff:ffff:ffff:ffff:ffff:ffff:ffff) A privately routable IPv6 unicast address used only for communication between nodes within your organization. The unique local addresses cannot be routed to the public internet. The address consists of a routing prefix in the fd00:/8 address spaces, assigned locally by your organization. And the full host address might be manually configured or automatically assigned using IPv6 auto-configuration and neighbor discovery.

Invalid IP Addresses for Discovery Seeds or Auto-Discovery Hints, continued

IPv4 Address Range	IPv6 Address Range	Explanation
not applicable	`fe80::*:*:*:*`	**IPv6 link-local address**[1]
`224-239.*.*.*`	not allowed (`ff00::` to `ffff:ffff:ffff:ffff:ffff:ffff:ffff`)	**multicast address**[2]
`255.255.255.255`	not applicable	Broadcast address

Communicate any additional IP address requirements to your team to avoid unexpected discovery results.

NNMi uses information gathered from Routers to establish membership for Subnet connections. Make sure that important Routers in your network environment are SNMP enabled.

NNMi uses either of the following criteria to identify a Router:

- The Router responds to an SNMP query with appropriate values for `sysServices` (1.3.6.1.2.1.1.7) and `ipForwarding` (1.3.6.1.2.1.4.1). See RFC 1213, MIB-II for details.
- The Router responds to an SNMP query with an appropriate MIB-II `sysObjectID` value according to the current settings in NNMi's Device Profile configuration.

You must provide the appropriate SNMP Community Strings to NNMi. See "Configuring Communication Protocol" on page 119.

Related Topics

"Discovery Seed Results" on page 270

"Delete Discovery Seeds" on page 280

Examine Discovery Results

When verifying discovery, you can do any of the following tasks:

[1] A non-routable IPv6 unicast address only used for communication with other nodes on the same link (LAN or VLAN). Link local addresses cannot be used for communication that must be forwarded through a router. IPv6 auto-configuration automatically assigns a unique link local address in the fe80::/10 address space to each IPv6-enabled interface on a system.

[2] Used to identify a group of hosts joined into a group. IPv4 multicast addresses are in the range 224.0.0.0 to 239.255.255.255 and IPv6 multicast addresses have the prefix ff00::/8.

Check Initial Progress of Discovery

During initial NNMi discovery of your network, you can check Spiral Discovery's progress in the following ways:

- Click **Help** → **System Information** (for more information see Displaying NNMi System Information):
  - Navigate to the **Database** tab to find the real-time list of discovery's progress.
  - Navigate to the **State Poller** tab to see a report of the health of the State Poller Service.
- To see state of discovery for a node, see "Node Discovery State Check" below.
- NNMi administrators can use the command line on any NNMi management server to generate a report about NNMi health. See the nnmhealth.ovpl Reference Page for more information.

Check this several times during a one hour period. The numbers in the Nodes, SNMP agents, Interfaces, IP addresses, and Layer 2 Connections fields stabilize when initial discovery is complete

> **Note:** If you configure one or more Auto-Discovery Rules and you get unexpected results, check your ordering numbers. See "Configure Auto-Discovery Rules" on page 215 for more information.

Node Discovery State Check

You can verify the current discovery state for a node.

To see the current Discovery State for a node:

1. Navigate to a **Node** form.
   a. From the workspaces navigation panel, select the workspace of interest. For example, **Inventory**.
   b. Select the node view of interest. For example **Nodes**.
   c. Select the row representing the configuration you want to see.
2. Locate the **Discovery State** attribute (in the Discovery section on the left side of the form).

   Possible values include:

   - **Newly Created** – Indicates the node and its IP addresses are in the NNMi database, but further information needs to be collected before state and status are determined.

- **Discovery Completed** – Indicates that discovery gathered all required information for the node.
- **Rediscovery in Process** – Indicates discovery is updating the information collected for the node.

Verify Success of Discovery Seeds

The discovery seeds provide the starting point for discovery.

To verify that each discovery seed was successfully discovered:

1. Navigate to the **Seeds** view.
   - From the workspace navigation panel, select the **Configuration** workspace.
   - Expand **Discovery**.
   - Select **Seeds**.
2. Check the value in the Discovery Seed Results column on each row of the table. A value of **Node Created** indicates the successful discovery of each discovery seed. See "Discovery Seed Results" below for the meaning of other values and how to correct discovery problems.

Discovery Seed Results

When you add a discovery seed, the Discovery Service immediately tries to discover it (without waiting until the next regularly scheduled discovery interval). If discovery is not successful, NNMi tries again 10 minutes later, and continues trying. The time between each try is doubled until it reaches 1 week or equals your current discovery interval.

To see the current discovery results for each specified discovery seed:

1. Navigate to the **Seeds** view
   - From the workspace navigation panel, select the **Configuration** workspace.
   - Expand **Discovery**.
   - Select **Seeds**.
2. The table lists each discovery seed and the result that NNMi gathered from the discovery seed. Check the value in the **Discovery Seed Results** column on each row of the table.

Discovery Seed Results Values

Discovery Results	Description
New seed	You just entered a new discovery seed. When discovery begins, Discovery Results changes to "In progress". If the "New seed" value does not change, check to see if the Discovery Service needs to be restarted, see "Verify that NNMi Services are Running" on page 85.
In progress	Discovery is in progress.
Node	The discovery seed is successfully discovered and a new Node is created in the

Discovery Seed Results Values, continued

Discovery Results	Description
created	database. When NNMi first discovers a seeded node, the *seed address* (provided by the NNMi administrator) is used for initial SNMP/ICMP communication. After NNMi builds an inventory of all IP addresses associated with the node (see "What Information Is Collected?" on page 177), NNMi follows a set of rules to determine which address is the best choice for each node's Management Address (see "Configure Default SNMP, Management Address, and ICMP Settings" on page 120). NNMi then uses the Management Address for all communication with the node.

Node created (non-SNMP device)	The hostname or IP address you provided is a non-SNMP device. The Node was discovered and added to the database, but no SNMP information is available because no SNMP agent responded.
	If this result is unexpected, the device might currently be down. Initiate an on-demand discovery poll using **Actions** → **Polling** → **Configuration Poll**
	Click here for more information. Or try the following:
	Tip: You can right-click any object in a table or map view to access the **Actions** menu.
	Check whether the IP address is accessible
	1. Type the following command to verify that the address is accessible: `ping <nodename>`
	Check the Access Control List
	1. Access the Node, and open the Access Control List (ACL).
	2. Verify that the NNMi management server address is in the list.
	Ensure that SNMP is working
	1. Use the nnmsnmpwalk.ovpl command. Type the following to verify that the address has an SNMP agent. Supply one specific MIB variable to limit network traffic to one object rather than requesting all possible SNMP values. For example, use the VendorID prefix:
	SNMPv1 or SNMPv2c:
	`nnmsnmpwalk -c <communityString> <nodename or IP address> <VendorID>`
	SNMPv3:
	`nnmsnmpwalk -c <v3u> <UserName> <VendorID>`
	2. If the nnmsnmpwalk.ovpl fails:
	a. Use telnet to check the device's SNMP configuration to verify that SNMP is enabled.

Discovery Seed Results Values, continued

Discovery Results	Description
	b. Verify that the address of the NNMi management server is listed in the SNMP Agent's Access list. **Check your communication configuration** 1. Verify that SNMP communication is enabled for this device: "Configuring Communication Protocol" on page 119. 2. Verify that the device has a properly configured SNMPv1 or SNMPv2c *read community string*, or that the device has a properly configured SNMPv3 USM security setting. 3. After you correct the problem that caused NNMi to specify the seed as a non-SNMP device, NNMi updates the Node record during the next discovery cycle. **Note:** The Discovery Results value does not change, because NNMi makes only one attempt to contact each discovery seed. However, everything is working properly once the Communication Configuration settings are corrected.
Node not created (DNS name resolution failed)	The Domain Name System (DNS) protocol could not match the hostname you provided for this discovery seed with a valid IP address.
Node not created (duplicate seed)	The address or hostname you provided is a Node that already exists in the database.
Node not created (IPv6 disabled)	The address you provided is an IPv6 address. NNMi Advanced is required, and the IPv6 feature must be enabled. The hostname you provided has only IPv6 addresses. NNMi Advanced is required, and the IPv6 feature must be enabled.
Node not created (IPv6 link local address is	The address you provided is an **IPv6 link-local address**[1], or the hostname you provided has only one address (an IPv6 link-local address). IPv6 link-local addresses cannot be used as seeds.

[1] A non-routable IPv6 unicast address only used for communication with other nodes on the same link (LAN or VLAN). Link local addresses cannot be used for communication that must be forwarded through a router. IPv6 auto-configuration automatically assigns a unique link local address in the fe80::/10 address space to each IPv6-enabled interface on a system.

Discovery Seed Results Values, continued

Discovery Results	Description
invalid seed)	
Node not created (license exceeded)	Discovery rejected this discovery seed because the number of devices previously discovered reached your licensed capacity limit. See "Extend a Licensed Capacity" on page 1575.
Failed	Contact with this discovery seed failed due to an internal NNMi error. The problem might be related to discovery or to a system wide issue, such as running out of memory or having trouble with database access. Check the discovery log file (see "Verify that NNMi Services are Running" on page 85): • **Windows:** `%NnmDataDir%\log\nnm\nnm.0.0.log` • **UNIX:** `/var/opt/OV/log/nnm/nnm.0.0.log`

Related Topics:

"Specify Discovery Seeds" on page 256

Examine Discovery Inventory

The best method for examining your discovered inventory depends on how you configure discovery.

To examine your Discovery Inventory:

1. In the **Workspace** navigation panel, open the **Inventory** workspace.
2. Select the **Nodes** view.
3. Verify that each important Node is listed.
4. Select the **IP Addresses** view.
5. Verify that each IP address that you identified as a discovery seed is listed.
6. Verify that the IP addresses you expect to see are visible (based on any Auto-Discovery Rule configurations - see "Configure Discovery " on page 199 and "Configure an Excluded IP Addresses Filter" on page 248).
7. To check on the current discovery state for a particular node, see "Node Discovery State Check" on page 269.

Note: If you configure one or more Auto-Discovery Rules and you get unexpected results, check your ordering numbers. See "Configure Basic Settings for the Auto-Discovery Rule" on page 218 for more information.

Related Topics

Using the IP Addresses View

Using the Nodes View

Examine Layer 2 Discovery Results

Layer 2 represents your network's physical connections and LAN switch traffic routes. For more information, see "Configure Tenants" on page 194.

Devices that belong to the Default Tenant can have Layer 2 Connections to any device in any Tenant. Devices within any Tenant *other than* Default Tenant can have Layer 2 Connections *only* to devices within the same Tenant or the Default Tenant.

> **Note:** A cloud icon on the NNMi map may represent a missing node. Consider using a router command such as Cisco `show cdp neighbors` to help identify those missing Nodes. Check the Access Control List (ACL) configurations in your network environment to fix the problems.

To examine Layer 2 inventory and connectivity results:

1. In the **Workspace** navigation panel, open the **Inventory** workspace.
2. Select the **Nodes** view.
3. Select the row representing the node of interest.
4. Select **Actions** → **Layer 2 Neighbor View**.

   > **Tip:** You can right-click any object in a table or map view to access the **Actions** menu.

5. Use the **Number of Hops** field to expand the area shown on the map.

   Number of Hops: 1

6. Examine your network connectivity to ensure it is as expected. See "Add or Delete a Layer 2 Connection" on page 284 if changes are required.

   > **Note:** If you configure one or more Auto-Discovery Rules and you get unexpected results:
   >
   > - Check your ordering numbers. See "Configure Basic Settings for the Auto-Discovery Rule" on page 218 for more information.
   > - Check the Layer 2 protocol configuration at each end of the problem connection. See "Troubleshooting Layer 2 Connections" on the next page.
   > - Check each Node's assignment for Tenant. The Tenant assignment can be easily changed, see "Change Tenant Assignment for a Node" on page 289. Subnets are calculated independently within each Tenant.

To examine VLAN results:

1. In the **Workspace** navigation panel, open the **Inventory** workspace.
2. Select the **VLANs** view.
3. Double-click the row representing the VLAN of interest.
4. Verify that the list includes all nodes and ports assigned to this VLAN.

> **Note:** If you configure one or more Auto-Discovery Rules and you get unexpected results, check your ordering numbers. See "Configure Basic Settings for the Auto-Discovery Rule" on page 218 for more information.

To examine Router Redundancy Group results:

1. In the **Workspace** navigation panel, open the **Inventory** workspace.
2. Select the **Router Redundancy Groups** view.
3. Use the Tenant assignment column to Sort the view, and verify that all members of each group are assigned to the same Tenant.
4. To correct any problems, change the Tenant assignments, see "Change Tenant Assignment for a Node" on page 289.

Related Topics

Using the Layer 2 Neighbor View

Using the Layer 3 Neighbor View

Troubleshooting Layer 2 Connections

If you get unexpected results for Layer 2 Connections in your network environment, review the following information.

A network device's interfaces can be configured with proprietary Layer 2 *discovery protocols*, instead of or in addition to the industry standard LLDP (see the list of Topology Source protocols in Layer 2 Connection Form).

By default, NNMi checks the interface for standard LLDP and vendor-specific IEEE 802 Layer 2 protocol. NNMi uses data from both protocols to calculate the Layer 2 Connection, but by default prefers the data provided through LLDP.

> **Note:** Forwarding Database (FDB) information can cause NNMi to establish wrong Layer 2 Connections in the following cases:
>
> - When the FDB is configured as cache and contains obsolete data.
> - In network environments with hardware from a variety of vendors, when each vendor generates different and sometimes conflicting FDB data.
>
> *Optional*: NNMi administrators can configure Spiral Discovery to ignore the FDB data from one Node Group when calculating Layer 2 Connections (the FDB data is still included in other calculations).

(*NNMi Advanced - Global Network Management feature*) NNMi must read the Forwarding Database (FDB) tables from Ethernet switches within the network before accurate communication paths between these network devices can be calculated. Because the FDB data is involved, NNMi can produce different results on a Regional Manager as opposed to the Global Manager.

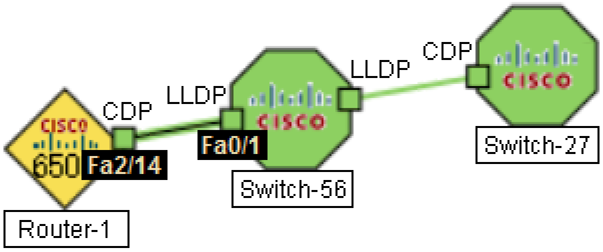
If NNMi discovers more than one IEEE 802 Layer 2 protocol being used by a particular device's interface, the Device Profile's setting controls NNMi's protocol preference:

☑ **Prefer LLDP** = Enabled: NNMi gives priority to the LLDP data.

□ **Prefer LLDP** = Disabled: NNMi gives priority to the vendor-specific IEEE 802 Layer 2 protocol data.

NNMi cannot detect accurate Layer 2 Connections under the following circumstances:

NNMi does not support the following scenario. Switch-56 has interfaces with one Layer 2 protocol enabled. The devices at the other end of Switch-56's Layer 2 Connections have a different Layer 2 protocol enabled:



NNMi detects a false connection directly from Router-1 to Switch-27.

To fix the problem, configure both sides of each Layer 2 Connection exactly the same (both interfaces enable either the same protocol or dual protocols).

See also "Ignore Forwarding Database Information from a Node Group" on page 208.

Examine Layer 3 Discovery Results

Layer 3 represents your network's router traffic.

To examine Layer 3 inventory results:

1. In the **Workspace** navigation panel, open the **Inventory** workspace.
2. Select the **Nodes** view.
3. Select the row representing the router of interest.
4. Select **Actions** → **Layer 3 Neighbor View**.

   Tip: You can right-click any object in a table or map view to access the **Actions** menu.

5. Use the **Number of Hops** field to expand the area shown on the map.

Number of Hops: 1

6. Examine your network connectivity to ensure it is as expected. If changes are required, try the following:

   - Use **Actions** → **Polling** → **Configuration Poll**. See Using Actions to Perform Tasks for more information.

   - Manually add or delete the connection. See "Add or Delete a Layer 2 Connection" on page 284 .

   - Verify that the addresses on each end of the connection are not listed in the Excluded IP Address filter. See "Configure an Excluded IP Addresses Filter" on page 248.

> **Note:** If you configure one or more Auto-Discovery Rules and you get unexpected results, check your ordering number for each rule. See "Configure Auto-Discovery Rules" on page 215 for more information.

Related Topics

Using the Layer 2 Neighbor View

Using the Layer 3 Neighbor View

Keep Your Topology Accurate

For suggestions see the following topics:

With NNMi, discovery is ongoing. After initial discovery, NNMi checks periodically to ensure that the maps accurately reflect the state of your network. By default, NNMi uses the following methods to keep network information accurate and up-to-date:

Spiral Discovery. NNMi tracks MAC addresses in addition to IP addresses so that NNMi knows when devices move from place to place in your network environment. See "What Information Is Collected?" on page 177.

Scheduled Rediscovery. Rediscovery occurs automatically at the interval you define. See "Configure Schedule Settings" on page 209 for more information about setting the discovery schedule.

***Optional:* Auto-Discovery (Default Tenant only)**. If you choose to use Auto-Discovery, NNMi uses information gathered from neighboring devices on your network to discover all devices connected to your network. See "Configure Basic Settings for the Auto-Discovery Rule" on page 218.

Delete Nodes

> **Tip:** To configure NNMi to automatically delete unresponsive nodes, see "Configure Whether to Delete Unresponsive Nodes" on page 212.
>
> To ensure that NNMi never discovers a particular Node in the future, change the Communication Configuration settings, see "Configuring Communication Protocol" on page 119.

Sometimes it is useful to delete Nodes. For example:

- Remove any nodes that are no longer being used in the network.
- Avoid reaching the NNMi license limit for number of managed Nodes by deleting less important Nodes.
- When non-SNMP addresses that had the same DNS hostname are changed to have separate DNS hostnames, NNMi must completely rediscover the non-SNMP nodes to correctly update the database objects (for example, node, interface, address, connection, and incidents).

> **Note:** If you delete a Node with many interfaces and VLANs, you might see an error message indicating that the Node could not be deleted. This means the database was busy with discovery. Try again between discovery cycles.

If a deleted Node is one of your seeds, delete that seed from the Discovery Seeds table as well. See "Delete Discovery Seeds" on page 280.

To understand the results of deleting a Node, click here for more information.

- NNMi cleans up the database by deleting the following objects:
  - Any objects representing a component of the deleted Node (for example, all of that node's interfaces and IP addresses).
  - Any related objects that are empty after deleting the Node (for example, subnets).
  - Any connections with only zero or one end points after deleting the Node.
  - The History of the Node object and all related objects.
- The time required for NNMi to finish deleting depends on the number of objects or related objects being deleted.
- During future discovery cycles, if the deleted Node meets the criteria for an Auto-Discovery Rule and appears in a monitored router's ARP cache, NNMi adds the Node back into the NNMi database during the next discovery cycle. To prevent this, create an Excluded IP Addresses filter for the addresses (see "Configure an Excluded IP Addresses Filter" on page 248).
- During future monitoring cycles, NNMi polls only objects currently in the database.
- Each Incident associated with the deleted Node is modified in the following ways, but not deleted from the NNMi database:

- The **Status** attribute changes to **Closed**.
- The **Correlation Notes** indicate the deletion of the associated node, interface, or address.
- The **RCA State** attribute changes to **FALSE**.

> **Note:** Incidents generated from SNMP traps or NNM 6.x/7.x Events (received from the deleted Node) appear in the Incident views, but remain unresolved.

- If you are viewing a Node that has recently been deleted by another user, the deleted Node appears as a transparent icon on the map until the map is refreshed using the ⟳ **Refresh** icon. After **Refresh**, the deleted node is removed from the map. NNMi does not automatically refresh the connectivity or set of nodes in a map view, except on the **Initial Discovery Progress** and **Network Overview** maps.

A subset of NNMi users can delete nodes from a table view, map view, or Node form (depending on the assigned NNMi Role).

> **Note:** By default NNMi Administrators can delete nodes. NNMi Administrators can configure NNMi to permit User Accounts assigned to the NNMi Operator Level 2 User Group to delete nodes. See the *HP Network Node Manager i Software Deployment Reference* for more information (**Help** → **Documentation Library**). Search for "Delete Node".

To delete one or more nodes (maximum 20 at one time):

1. Unmanage the nodes you want to delete.
   a. In a table view, press CTRL-Click and select each row that represents a node you want to unmanage.
   b. Select **Actions** → **Management Mode** → **Unmanage**.

   > **Tip:** You can right-click any object in a table or map view to access the **Actions** menu.

   c. Wait until the Status=*No Status* for each of the following objects:
      - Each Node to be deleted
      - Each Node's Interfaces, IP Addresses, Cards, Ports, and VLAN Ports
2. Do one of the following:
   - *Table views*: Press CTRL-Click and select each row that represents the objects of interest, and click the ✖ Delete icon. Each selected node is deleted from the NNMi database and removed from the current view.
   - *Map views*: click the map symbol representing the node you want to delete, and click **File** → **Delete Node**. The node is deleted from the NNMi database and removed from the current view.
   - *Node form*: select **File** → **Delete Node** and in the confirmation dialog, click **OK**. The form is automatically closed after NNMi deletes the Node.

> **Note:** If the delete fails, use the nnmnodedelete.ovpl command. Wait for the command to complete.

To delete any number of nodes:

Use the `nnmnodedelete.ovpl` command. See the nnmnodedelete.ovpl Reference Page.

Related Topics

Using Table Views

Using Map Views

Delete Discovery Seeds

There are two ways to delete discovery seeds from the NNMi Discovery configuration and the NNMi database.

Note: If you remove a Discovery Seed from Discovery Configuration, the corresponding node is not deleted from the topology database. See "Delete Nodes" on page 1602 for information about removing the entire node record from the topology database.

To delete seeds using the Discovery Configuration view:

1. Navigate to the **Seeds** view.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Expand **Discovery**.
   c. Select **Seeds**.
2. To delete one or more discovery seeds, press CTRL-Click and select each row that represents a node you want to delete.
3. Click the ✖ Delete icon.

To delete any number of seeds at one time from the command line:

At the command line of the NNMi management server, type the nnmseeddelete.ovpl command.

Case-sensitive exactly as listed in the **Discovery Seeds** tab in the Discovery Configuration form, specify the hostname or IP address.

If you do not want to enter an NNMi User Name attribute value and an NNMi Password attribute value at the command line, you can use the nnmsetcmduserpw.ovpl command to specify the valid user name and password (instead of `-u` and `-p`). The credentials set using the `nnmsetcmduserpw.ovpl` command are valid for command execution by the same user. See "Set Up Command Line Access to NNMi" on page 577 for more information.

```
nnmseeddelete.ovpl -seed <hostname/IP-address> -u <NNMiadminUserName>
-p <NNMiadminPassword>
```

Detect Interface Changes

During each Spiral Discovery cycle, NNMi responds to Interface changes as follows:

1. NNMi updates the attribute value of the current Interface object if one (*and only one*) of the following attributes change:
   - `ifIndex` or `IfAlias` or `ifSpeed`
2. NNMi creates a new Interface object and deletes the old Interface object if any of the following criteria are met:
   a. At least one of these attributes change: `ifName`, `ifDescriptions`, `ifType`, or Physical Address (Mac address, Media Access Control address).

   b. More than one of these attributes change: `ifIndex` or `IfAlias` or `ifSpeed`.

   c. One or more attributes from the list of both criteria 1 & 2 change.

   > **Note:** If using nnmconnedit.ovpl configuration files, any connection settings configured for the deleted Interface would be evaluated for the new Interface object’s current attribute settings.

To troubleshoot interface changes in your network environment, do one of the following:

- For immediate results, navigate to a Node view and select one of the problem devices.

  Click **Actions** → **Polling** → **Configuration Poll** to instruct Spiral Discovery to rediscover the Node, updating information about interfaces within that device.

  > **Tip:** You can right-click any object in a table or map view to access the **Actions** menu.

  Open the Node form for the device and verify that the list of interfaces is correct.
- Wait until the next regularly scheduled Discovery or Monitoring cycle (controlled by the Interval settings the NNMi administrator specifies in the Discovery and Monitoring configuration forms).

NNMi administrators use multiple configuration settings to control how NNMi detects interface changes. To troubleshoot issues, verify the current settings for the following:

1. *Prerequisite:* To detect interface changes, the **Configuration** > **Monitoring** > ☑ **Enable Interface Fault Polling** must be enabled. This setting is available at three levels (see "Configure NNMi Monitoring Behavior" on page 340 for more information):
   - Interface Group Settings tab > Interface Group Settings form > Fault Monitoring section
   - Node Group Settings tab > Node Group Settings form > Fault Monitoring section
   - Default Settings tab: Default Fault Monitoring section

   If enabled and the NNMi State Poller detects a change, NNMi does the following:
   - Generates a request for Spiral Discovery to rediscover the Node (checking for any changes). If NNMi is busy gathering other information, it may take a while for this request to get to the top of the queue. If NNMi is not busy, the results might seem immediate.
   - Suspends monitoring of that node until NNMi finishes gathering the updated information about the Node itself (or for 30 minutes maximum).
2. **Configuration** > **Monitoring** > Monitoring Configuration's Default Settings tab, Default

Change Detection Monitoring block of attributes. If **Number of Interfaces (ifNumber) Polling** ☑ is enabled, NNMi does the following:

- Polls for the *total number of interfaces* within the Node by requesting an SNMP response to MIB II `ifNumber`.
- NNMi compares the answer for *total number of interfaces* within the Node, to the previous answer from that Node's SNMP agent.
- If the number has changed, Spiral Discovery redisovers the Node.
- NNMi suspends fault, performance, and status monitoring of that Node until updated information about hardware is gathered.

See "Default Settings for Monitoring" on page 345 and "Node Group Settings for Monitoring" on page 391 for more information.

> **Tip:** This setting detects whether the *total number of interfaces* within the node has increased or decreased. To detect whether the actual number assigned to particular interfaces has changed (the `ifIndex` value), continue with the next step.

3. For each node vendor/make/model (RFC 1213, MIB-II, `sysObjectID`), the NNMi administrator chooses which interface MIB variable the NNMi State Poller queries to detect interface changes.

   Configuration > **Device Profiles**: On the Advanced tab, the **Interface Reindexing Types** attribute instructs NNMi to do the following (see Device Profile Form for more information about the four SNMP values involved in this calculation).

Interface Reindexing Types

MIB II Variable Used to Detect a Change	How State Poller Detect Changes
`ifIndex` value **Note: Note**: Use `ifIndex` only for manufacturers/models that maintain a static `ifIndex` list.	If an SNMP agent's previous response for SNMP `ifIndex` values (numbers assigned to each interface) does not match the current response, State Poller requests that NNMi gather new information about the interfaces within the Node. For example, someone installs or removes interfaces from a device in your network: ▪ Use this MIB-II `IfIndex` setting for devices that maintain a static list of MIB-II `IfIndex` numbers. ○ When interfaces are added - MIB-II `IfIndex` numbers are added to the end of the current list of interfaces contained in that device. ○ When interfaces are removed - the MIB-II `IfIndex` numbers previously used by those interfaces are dropped from the list. ▪ Do not use this MIB-II `IfIndex` setting for devices that reset all MIB-II `IfIndex` numbers for the group of interfaces

Interface Reindexing Types , continued

MIB II Variable Used to Detect a Change	How State Poller Detect Changes
	contained in that device each time a change occurs. Each manufacturer has a different strategy for identifying each interface and detecting when an existing interface is simply assigned to a different MIB-II `IfIndex` number or an interface is removed. **Caution:** When you choose `ifIndex`, NNMi can detect when a particular number no longer exists (static assignments). However, this choice might not detect interface renumbering (a value now being used by a different interface). To detect this type of interface renumbering, choose any combination of `ifName` and `ifDescr` and `ifAlias` settings, below.
`ifName` value	Based on the `ifIndex` number, compares the `ifName` value on the interface with the previously discovered `ifName` value. If changes in this name/number relationship are detected, State Poller requests NNMi to gather new information about the Node's interfaces.
`ifDescr` value	Based on the `ifIndex` number, compares the `ifDescr` value on the interface with the previously discovered `ifDescr` value. If changes in this description/number relationship are detected, State Poller requests NNMi to gather new information about the Node's interfaces.
`ifAlias` value	Based on the `ifIndex` number, compares the `ifAlias` value on the interface with the previously discovered `ifAlias` value. If changes in this alias/number relationship are detected, State Poller requests NNMi to gather new information about the Node's interfaces.
Combination of `ifName` or `ifDescr` values	Based on the `ifIndex` number, compares the `ifDescr` and `ifName` values on the interface with the previously discovered values. If changes are detected, State Poller requests NNMi to gather new information about the Node's interfaces.
Combination of `ifName` or `ifDescr` or `ifAlias` values	Based on the `ifIndex` number, compares the `ifName` and `ifDescr` and `ifAlias` values on the interface with the previously discovered values. If changes are detected, State Poller requests NNMi to gather new information about the Node's interfaces.

Tip: Open any Node form and navigate to the Basics' **Device Profile** link. You can open the associated Device Profile to see the current setting.

4. The next time each Node is rediscovered, if something has changed, Spiral Discovery compares the current `ifIndex` value against the MAC address to determine whether an interface was added, deleted, or renumbered.

Add or Delete a Layer 2 Connection

Layer 2 Connections are only permitted between the Default Tenant, and other Tenants (never between two non-Default Tenants). For more information, see "Configure Tenants" on page 194.

If your network management domain includes ATM, Frame Relay, or **MPLS**[1] links between wide area networks (WANs), you might need to use the connection editor to show the links in the Layer 2 Neighbor View maps within NNMi. For MPLS, you can provide multiple connections between two nodes.

See also the Schedule Settings for Spiral Discovery: "Configure Whether to Delete Unresponsive Nodes" on page 212 and "Configure Whether to Delete Layer 2 Connections" on page 213.

Subnet Connection Rules

Subnet Connection Rules are ideal for multiple situations. See "Consider IPv4 Subnet Connection Rules" on page 181 for more information.

NNMi uses Subnet Connection Rules to detect connections between interfaces associated with IPv4 addresses that *do not respond* to Layer 2 *discovery protocols* (see the list of Topology Source protocols in Layer 2 Connection Form). Subnet Connection Rules take priority over the Layer 2 discovery protocol results. For special cases, you can override a Subnet Connection Rule by using the Connection Editor command line tool, see nnmconnedit.ovpl for more information.

NNMi provides a variety of predefined Subnet Connection Rules. For ideas, see "Subnet Connection Rules Provided by NNMi" on page 247.

Connection Editor (to add or delete connections)

In the Inventory workspace > Layer 2 Connections view, you can see a list of connections. No Delete action is available in the Layer 2 Connections view.

Use the nnmconnedit.ovpl command line tool to do the following:

- delete a connection data
- add connection data
- instruct NNMi to ignore certain connection data

The `nnmconnedit.ovpl` command is used to generate a template XML file (shown in the following example). For each connection to be added or deleted, you provide information about the node and interface at both ends of the connection. Multiple `<connection>` elements are permitted within the template XML file.

```
<connectionedits>
 <connection>
    <operation>add or delete</operation>
    <node>node Name, Hostname or management IP address</node>
    <interface>ifName, ifAlias, ifDescr or ifIndex</interface>
    <node>node Name, Hostname, or management IP address</node>
```

[1] Multiprotocol Label Switching

```
  <interface>ifName, ifAlias, ifDescr or ifIndex</interface>
 </connection>
</connectionedits>
```

To add or delete a connection:

1. For the devices at both ends of the connection, gather the data required to identify the device and interface (see table).
2. On the NNMi management server, at the command line, generate a connections template file using either `add` to create an add.xml template file or `delete` to create a delete.xml template file.

   In the following example, NNMi creates an add.xml file:
   `nnmconnedit.ovpl -t add`

   Note: If you specify add, NNMi creates the template file named `add.xml`. If you use delete, the template file is named `delete.xml`.
3. Open the template file in a text editor and fill in the correct information for each node and interface.
4. On the NNMi management server, at the command line, load the new connection information into the NNMi database:
   `nnmconnedit.ovpl -f <add|delete>.xml`

   For example, to load the add.xml template file, enter:

   `nnmconnedit.ovpl -f add.xml`
5. Open the Layer 2 Neighbor View map and verify the connection changes.

Required Layer 2 Connection Attributes in the Connection Editor File

Attribute	Description
operation	Specify whether the connection is to be added or deleted.
node	Identify the node using any of the following *case-sensitive* values: • `node Name` • `Hostname` (*case-sensitive*) NNMi follows a set of rules to dynamically generate the value stored in the NNMi database for each Node's Hostname. Click here for details. ▪ If the Node supports SNMP, NNMi requests the Hostname using the IP Address of the associated SNMP agent (the Management Address attribute value on the Node form). When the NNMi administrator chooses **Enable SNMP Address Rediscovery** ☑ in the Communication Configuration: ○ If the SNMP Agent does not respond, NNMi checks for another Management Address to request the Hostname, and the Hostname could change. ○ If the SNMP Agent associated with the node changes, the Management Address and Hostname could change.

Required Layer 2 Connection Attributes in the Connection Editor File , continued

Attribute	Description
	When the NNMi administrator disables **Enable SNMP Address Rediscovery** ☐ in the Communication Configuration: ○ If the SNMP Agent does not respond, NNMi uses the previously gathered Management Address attribute value to request the Hostname. ○ If the SNMP Agent associated with the node changes, NNMi uses the previously gathered Management Address attribute value to request the Hostname. ■ If the Node does not support SNMP, no Management Address is available. NNMi requests a Hostname starting with the lowest IP Address associated with the node (a Discovery Seed value or an IP address value gathered from a neighboring device). NNMi uses the first Hostname provided. The Hostname might change during a future discovery cycle. **Note:** NNMi administrators can use NNMi property file settings to change the way NNMi determines Hostname values: ■ `nms-topology.properties` file settings: If DNS is the source of the Node's Hostname, there are three choices. By default NNMi uses the exact Hostname from your network configuration. It is possible to change NNMi behavior to convert Hostnames to all uppercase or all lowercase. See the "Modifying NNMi Normalization Properties" section of the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`. ■ `nms-disco.properties` file settings: The Hostname is either requested from the Node's lowest loopback interface IP address that resolves to a Hostname or requested from the Node's designated Management Address (SNMP agent address). With either choice, when no IP address resolves to a Hostname, the IP address itself becomes the Hostname. See the "Maintaining NNMi" chapter of the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`. • `management IP address` NNMi follows a set of rules to determine which address is the best choice as the node's Management Address. Click here for details. **Note:** With NNMi Advanced, the NNMi administrator specifies whether NNMi prefers IPv4 or IPv6 addresses when selecting the Management Address. See Configure Default SNMP, Management Address, and ICMP Settings. a. NNMi ignores the following addresses when determining which Management Address is most appropriate:

Required Layer 2 Connection Attributes in the Connection Editor File , continued

Attribute	Description
	○ Any address of an administratively-down interface. ○ Any address that is virtual (for example, **VRRP**[1]). ○ Any IPv4 **Anycast Rendezvous Point IP Address**[2] or IPv6 Anycast address. ○ Any address in the reserved loopback network range. IPv4 uses 127/24 (`127.*.*.*`) and IPv6 uses `::1`. ○ Any **IPv6 link-local address**[3]. b. If the NNMi Administrator specifies Enable SNMP Address Rediscovery in Communication Configuration, NNMi prefers the last-known Management Address (if any). c. If the Management Address does not respond and the NNMi Administrator specifies Enable SNMP Address Rediscovery in Communication Configuration, NNMi uses the Communication Configuration settings for *Management Address Selection*. The NNMi Administrators chooses the order in which NNMi checks the following: ○ Seed IP / Management IP - If the NNMi Administrator configures a Seed, NNMi uses the Seed address (either a specified IP address or the DNS address associated with a specified hostname) only during initial Discovery. NNMi then requests the current Management Address (the address from which the node's SNMP Agent responds) and uses that IP address for all communication after initial discovery. ○ Lowest Loopback - If a node supports multiple **loopback address**[4], NNMi queries each loopback addresses, starting with the lowest number. NNMi uses the loopback address with the lowest number from which the SNMP agent responds (for example, 10.16.42.197 is a lower number than 10.16.197.42).

[1] Virtual Router Redundancy Protocol

[2] Rendezvous Point addresses are loopback addresses used for routers in multi-cast network configurations.

[3] A non-routable IPv6 unicast address only used for communication with other nodes on the same link (LAN or VLAN). Link local addresses cannot be used for communication that must be forwarded through a router. IPv6 auto-configuration automatically assigns a unique link local address in the fe80::/10 address space to each IPv6-enabled interface on a system.

[4] The address associated with the loopback interface. The loopback interface is a virtual interface on a device that provides a route for internal communication. Many vendors provide a specially configured loopback for management purposes. Exact details of how loopbacks are configured varies by vendor and model. See each device's documentation for details. NNMi identifies these loopback addresses by using ifType Number 24, softwareloopback from the IANA ifType-MIB.

Required Layer 2 Connection Attributes in the Connection Editor File , continued

Attribute	Description
	○ Highest Loopback - If a node supports multiple loopback address[1], NNMi queries each loopback addresses, starting with the highest number. NNMi uses the loopback address with the highest number from which the SNMP agent responds. ○ Interface Matching - The NNMi Administrator chooses which interface MIB variable NNMi queries to detect changes. NNMi can use the following MIB-II attribute values: `ifIndex`, `ifName`, `ifDescr`, `ifAlias`, or a combination of these (`ifName` or `ifDescr`, `ifName` or `ifDescr` or `ifAlias`). d. If no response, NNMi queries any remaining IP addresses in the node's IP address inventory, starting with the lowest number. NNMi uses the address with the lowest number from which the SNMP agent responds. e. If no response, NNMi checks for any Mapped Address configured for one of the currently known addresses (see the Mapped Address column in the Custom IP Addresses view). **Note:** The address represents a *static* Network Address Translation (NAT) pair's *external IP address* from the internal/external IP address pair. NNMi Administrators configure these pairs using the Overlapping IP Address Mapping form. NNMi uses this list of addresses starting with IPv4 from low to high, then IPv6 from low to high. f. If no response, NNMi might be configured to repeat the sequence using SNMPv1, SNMPv2c, or SNMPv3 in the order specified by the NNMi administrator (Communication Configurations *SNMP Minimum Security Level* settings). g. When all else fails, NNMi retains the last known Management Address (if any) and automatically changes the State of that SNMP Agent object to Critical. This process is repeated during each Spiral Discovery cycle, and the Management Address can change. For example, NNMi's inventory of addresses for the node expands, or the current Management Address does not respond to SNMP queries due to network problems or node reconfiguration. The NNMi administrator can prevent changes to the management address using the Communication Configurations *Enable SNMP Address Rediscovery* or *Preferred Management Address* setting.
interface	Identify the interface using one or more of the following (MIB-II) values:

[1] The address associated with the loopback interface. The loopback interface is a virtual interface on a device that provides a route for internal communication. Many vendors provide a specially configured loopback for management purposes. Exact details of how loopbacks are configured varies by vendor and model. See each device's documentation for details. NNMi identifies these loopback addresses by using ifType Number 24, softwareloopback from the IANA ifType-MIB.

Required Layer 2 Connection Attributes in the Connection Editor File , continued

Attribute	Description
	• `ifName` • `ifAlias` • `ifDescr` • `ifIndex` Note the following for `ifIndex`: ▪ For interfaces in Non-SNMP nodes, always use the `ifIndex` value of 0 (zero). ▪ For interfaces in SNMP nodes, choose other MIB-II values to identify the interface because often automatic interface renumbering causes confusion. See"Detect Interface Changes" on page 280.

Start Discovery On-Demand

NNMi provides the nnmnoderediscover.ovpl command line tool for initiating discovery. This tool enables NNMi administrators to do the following:

- Run discovery of a subset of your network domain to get the most recent data without waiting for the next-regularly schedules discovery cycle.

  For example: Use nnmnoderediscover.ovpl to immediately add newly deployed critical devices to the NNMi database without waiting for the next regularly-scheduled discovery cycle.

- Run discovery of your entire network on demand or using an automation script.
- Request updated discovery results from the Regional Managers in your network environment after restoring the Global Manager to a previous state.

  (*NNMi Advanced - Global Network Management feature*) Any change to the *Node's* Management Mode setting is immediately sent from a Regional Manager (NNMi management server) to the Global Manager. (Changes to Management Mode for other objects are sent during the next Spiral Discovery cycle on the Regional Manager.)

  Note: This tool can help you synchronize the Global Manager if for some reason the original information from the Regional Managers is lost from the Global Manager's database.

See nnmnoderediscover.ovpl for more details.

Change Tenant Assignment for a Node

After discovery, NNMi administrators can change the Tenant settings for any Node:

- Using the nnmsecurity.ovpl command to change multiple Nodes.
- Using the Node form to change one Node's setting.

Devices that belong to the Default Tenant can have Layer 2 Connections to any device in any Tenant. Devices within any Tenant *other than* Default Tenant can have Layer 2 Connections *only* to devices within the same Tenant or the Default Tenant.

> **Tip:** Assign any infrastructure device that interconnects multiple NAT domains (such as a NAT gateway) to the Default Tenant. This ensures that NNMi displays the Layer 2 Connections your team and customers need to see.

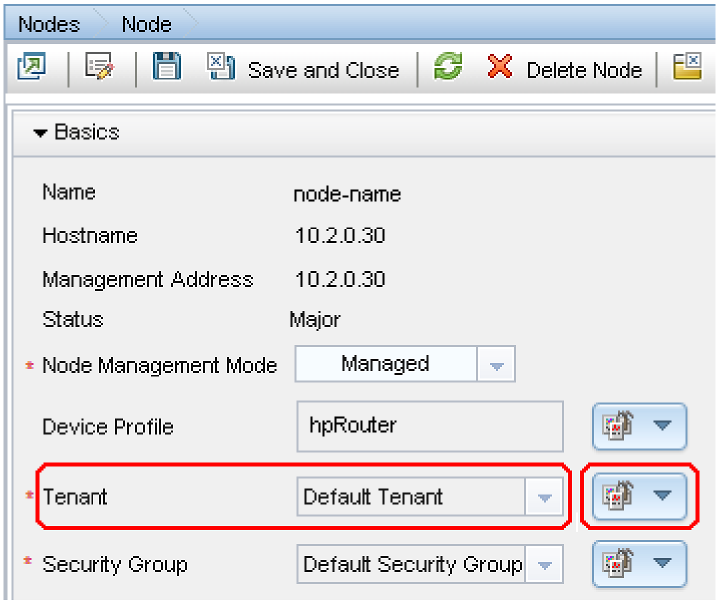
> **Caution:** Devices within the Default Security Group are visible from all views. To control access to a device, assign that device to a Security Group other than Default Security Group.

To assign a Node to a different Tenant:

1. Open the Node's form:

   > **Note:** Until an NNMi Administrator defines at least one Tenant in addition to Default Tenant (provided by NNMi):
   >
   > - The Tenant attribute does not appear on any Node form.
   > - The Tenant column does not appear in the Nodes (All Attributes) view.

2. In the Tenant attribute, do one of the following:



Name	node-name
Hostname	10.2.0.30
Management Address	10.2.0.30
Status	Major
* Node Management Mode	Managed
Device Profile	hpRouter
* Tenant	Default Tenant
* Security Group	Default Security Group

   - Select the drop-down list and choose a different Tenant.
   - Select the ▾ Lookup icon and select ✳ New to create a new Tenant.

3. Click **Save and Close**.

   > **Caution:**
   >
   > - If the Node is currently a member of a Router Redundancy Group, NNMi creates duplicate Nodes. You must manually delete the record of this Node that is associated with the prior Router Redundancy Group/Tenant pair.

> - If the Node was or is now participating in a *static* Network Address Translation domain, you must manually update any associated Overlapping IP Address Mapping. For more information:

4. *Optional*. Any seed configuration that assigned that Node to the old Tenant during initial discovery is now ignored by NNMi. Deleting the obsolete seed configuration is optional.

   a. Navigate to the **Seeds** view.

      i. From the workspace navigation panel, select the 🔧 **Configuration** workspace.

      ii. Expand **Discovery**.

      iii. Select **Seeds**.

   b. Select the row for the seed configuration that assigns that Node to the old Tenant, and click the ✖ Delete icon (see "Delete Discovery Seeds" on page 280 and "Delete Nodes" on page 1602 for more information).

Chapter 8

Configure Device Profiles

You can modify the settings in the Device Profiles to fine-tune Spiral Discovery and the device symbols on the maps.

According to industry standards (RFC 1213, MIB-II), each combination of vendor, device type, and model number is assigned a unique SNMP system object ID (`sysObjectID`). For example, all Cisco 6500 series switches have the same `sysObjectID` prefix: `.1.3.6.1.4.1.9.*`

HP provides well over three thousand preconfigured Device Profiles, one for each known `sysObjectID` at the time NNMi released.

NNMi uses Device Profiles (which equate to `sysObjectID`) to control certain types of behavior:

- Spiral Discovery determines the closest matching device profile, and uses the device profile settings to control certain attribute values for the discovered device. The Device Profile also influences the following:
  - Auto-Discovery Rules can provide an `sysObjectID` list that expandsthe default discovery behavior (beyond routers and switches) or prevents troublesome device types from being discovered.
  - The Node Name value might be affected, depending on your choices, see "Configure the Node Name Strategy" on page 203.
- When Node Groups are defined based on system object IDs, the State Poller Service monitors devices based on attribute values in the device profiles.
- Device Profile settings influence how State Poller detects renumbered interfaces. See "Detect Interface Changes" on page 280.
- In Map views, the background shape of map icons is determined by the Device Category. See About Map Symbols for an example of each available shape. There is also a Force Device attribute that enables category overrides in troublesome situations.

> **Tip:** To quickly locate the device profile settings for a particular network device, sort or filter the Device Profiles view by clicking the heading for the Device Vendor, Device Model, or Device Category columns.

To access the device profile definition for a particular device type:

1. Navigate to the **Device Profile** view.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Select the **Device Profiles** view.
2. Do one of the following:

- To create a device profile, click the * New icon.
- To edit a device profile, click the Open icon in the row representing the configuration you want to edit.

3. Modify the settings as needed:

   > **Caution:** When you make a change, NNMi must update all references to device profiles. This can take some time and slow down your system. Consider making this change during a slow time in your network environment.

   - The basic settings Device Category attribute value modifies NNMi behavior for Spiral Discovery and map symbols.

     > **Caution:** If you make changes to a Menu Item provided by NNMi, those changes are at risk of being overwritten in the future. See Author form for important information.

   - The advanced settings control NNMi behavior for Spiral Discovery and Node name selection. For example, instruct NNMi to treat a certain device type as a Router.

4. Click **Save and Close**. NNMi applies your changes during the next regularly scheduled discovery cycle. To apply the changes immediately, use **Actions** → **Polling** → **Configuration Poll**. See Using Actions to Perform Tasks for more information.

> **Tip:** You can right-click any object in a table or map view to access the **Actions** menu.

Chapter 9

Creating Groups of Nodes or Interfaces

Groups of nodes or interfaces are used for a variety of purposes within NNMi. Use of these groups is optional.

- Use node and interface groups to create custom view filters that help your team quickly sift through data in the NNMi views and identify the most important information. See Filter Views by Node or Interface Group.
- Special Actions are available for Node Groups and Interface Groups.
- Use Node Groups and Interface Groups to specify monitoring configuration settings. See"Monitoring Network Health" on page 340. For example, configure a different health monitoring interval for each group.
- (*NNMi Advanced* - Global Network Management *feature*) On a Regional Manager, use Node Groups to limit the amount of data available to Global Managers in your network environment. See "Regional Manager: Create a Forwarding Filter (Limit the available Node information)" on page 102 for more information.
- (*NNM iSPI Performance*) If you are using the HP Network Node Manager iSPI Performance for Metrics Software or HP Network Node Manager iSPI Performance for Traffic Software, control performance monitoring and provide report filters by Node Group.

  NNM iSPI Performance for Metrics only. NNMi automatically synchronizes Interface Group and Node Group configuration changes between NNMi and NNM iSPI Performance for Metrics. However, in some cases, additional configuration changes that affect Node Group or Interface Group membership might take longer to synchronize. If you do not see one or more nodes in an NNMi Performance for Metrics report that are visible in NNMi, use the **Actions** → **HP NNM iSPI Performance** → **Sync Interface and Node Groups** with NNMi option. This option forces NNMi to synchronize the Interface and Node Group information between NNMi and NNM iSPI Performance for Metrics more quickly than the default time frame.

Once Node Groups or Interface Groups are defined, you can reuse them within any context (view filtering and NNMi configuration settings) or you can configure them to be hidden from the view filter lists.

View Filter Possibilities

	Available in NNMi views based on: Object Type					
Filter	Incident	Node	Interface	IP Address	Card	Node Component
Node Groups "Create Node Groups" on the next page	x	x	x	x		
Interface Groups "Create Interface Groups" on page 321			x	x	x	x

Create Node Groups

Node Groups are used for a variety of purposes in NNMi. See "Creating Groups of Nodes or Interfaces" on the previous page for more information.

You can create any number of Node Groups in addition to the ones that NNMi provides (see "Node Groups Provided by NNMi" on page 335).

To create Node Groups, use one or more of the following methods:

- "Create Node Groups Using Filters or Hostname Lists (Configuration: Node Groups)" on the next page
- "Create Node Groups From the Actions Menu " on page 312
- "Add Nodes to a Node Group From the Actions Menu" on page 314
- "In a CSV File, Define Node Groups" on page 316

To verify the contents of the current Node Group:

1. In the Node Group form, click **Save**.
2. Select **Actions** > **Node Group Details** > **Preview Members (Current Group Only)**.
3. Click Refresh to check for the most recent changes to Node Group contents.

> **Tip:** To test the effects of your Node Group definition on Child Node Groups, in the Node Group form, select **Save**, then **Actions** > **Node Group Details** > **Show Members (Include Child Groups)**. NNMi displays the members of the current Node Group members as well as the members of each associated Child Node Group. Depending on the complexity of your Node Group hierarchy, NNMi might take some time to complete updating the results. Click Refresh to check for the most recent changes to Node Group contents.

Special Actions are available for Node Groups and Interface Groups.

> **Tip:** You can right-click any object in a table or map view to access the **Actions** menu.

You can also use the nnmloadnodegroups.ovpl command to list the following:

- Names of the existing Node Groups
- Selected attributes of nodes that are members of a specified Node Group

> **Tip:** *NNM iSPI Performance for Metrics only*. NNMi automatically synchronizes Interface Group and Node Group configuration changes between NNMi and NNM iSPI Performance for Metrics. However, in some cases, additional configuration changes that affect Node Group or Interface Group membership might take longer to synchronize. If you do not see one or more nodes in an NNMi Performance for Metrics report that are visible in NNMi, use the **Actions** → **HP NNM iSPI Performance** → **Sync Interface and Node Groups** with NNMi option. This option forces NNMi to synchronize the Interface and Node Group information between NNMi and NNM iSPI Performance for Metrics more quickly than the default time frame.

NNMi administrators can use Security Groups as Node Group definitions that become filters in NNMi views. If a user cannot access any nodes in a particular Node Group, that filter dynamically disappears from the filter selection list in the user's NNMi views. Any attribute in a Node form can be used to identify the members of a Node Group (for example, the Security Group attribute value or the Tenant attribute value).

> **Note:** If you use multiple tenants, you might not want users to see all of the Node Groups you create. To remove the Nodes Group view from the NNMi console, see the "NNMi Console" chapter of the *HP Network Node Manager i Software Deployment Reference*.

Related Topics

"Define Node Group Map Settings" on page 488

"Create Interface Groups" on page 321

Create Node Groups Using Filters or Hostname Lists (Configuration: Node Groups)

Node Groups are used for a variety of purposes in NNMi. See "Creating Groups of Nodes or Interfaces" on page 294 for more information.

> **Note:** By default NNMi Administrators can create, modify, and delete Node Groups. NNMi Administrators can configure NNMi to permit User Accounts assigned to the NNMi Operator Level 2 User Group to create, modify, and delete Node Groups. See the *HP Network Node Manager i Software Deployment Reference* for more information (**Help** → **Documentation Library**). Search for "Node Group".

You can create any number of Node Groups in addition to the ones that NNMi provides (see "Node Groups Provided by NNMi" on page 335).

One method for creating Node Groups is using filters or hostname lists to match the way your team identifies important network devices. Each Node Group is defined using one or more of the following:

- Device Filters (by any combination of SNMP device category, vendor, family, profile)
- Additional Filters (Boolean expressions based on a list of object attributes)
- Additional Nodes (identified by *case-sensitive* Hostname)
- Child Node Groups (use any combination of Node Groups to create a filter)

NNMi combines the results of all Node Group configuration settings in the following manner:

- NNMi first evaluates Device Filters. If any exist, nodes must match *at least one* specification to belong to this Node Group.
- NNMi then evaluates any Additional Filters. Nodes *must also pass all* Additional Filters specifications to belong to this Node Group.

- Any Additional Nodes specified are *always* included in the Node Group, regardless of any filters.
- Any Child Node Group results are treated the same as Additional Nodes.

> **Note:** You can also create Node Groups using the **Actions** → **Node Group Membership** option. This method adds the selected nodes to a Node Group that NNMi creates. See "Create Node Groups From the Actions Menu " on page 312 for more information.

To create a Node Group Using Filters or Hostname Lists (if your role permits you to do this):

1. Navigate to the **Node Group** form.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Expand the **Object Groups** folder.
   c. Select the **Node Groups** view.
   d. Do one of the following:
      - To create a Node Group, click the ✱ New icon.
      - To edit a Node Group, click the Open icon in the row representing the Node Group you want to edit.
2. In the Node Group form, provide the required information in the Basics section.
3. (*NNM iSPI Performance*) Make the Node Group available within NNM iSPI Performance products (see NNM NNM iSPI Performance table).
4. Identify the nodes that belong to this Node Group.

   Do one or more of the following:

   - Specify a filter based on Device Profile settings using the Device Filters tab (any combination of category, vendor, family, or profile).

     > **Tip:** To base your filter on the SNMP system Object ID number, use the Additional Filters `sysOidNode` code.

   - Specify a Node Group filter using the Additional Filters tab (use a variety of available codes to filter by object attribute values in the NNMi database).
   - Specify individual nodes using the Additional Nodes tab (provide a list of Hostnames, as they appear in the NNMi database).
   - Specify Child Node Groups using the Child Node Groups tab (use combinations of Node Groups to create a filter).
5. Click **Save and Close** to return to the Node Group form.

   > **Note:** You must click **Save and Close** to save your changes each time you create a Node Group.

6. Click **Save and Close**.

If you configured this Node Group for Monitoring, NNMi applies your changes during the next monitoring cycle. "Configure NNMi Monitoring Behavior" on page 340.

To review a Node Group definition:

1. From the workspace navigation panel, select the **Inventory** workspace.
2. Select the **Node Groups** view.
3. Double-click the row representing the Node Group definition you want to see.
4. The Node Group form displays.

   > **Note:** NNMi monitors the status of each Node Group over time. To check Node Group status information, access the Node Group form's Status tab.

5. When finished, click the Close icon.

You can also use the nnmloadnodegroups.ovpl command to list the following:

- Names of the existing Node Groups
- Selected attributes of nodes that are members of a specified Node Group

Special Actions are available for Node Groups and Interface Groups.

Related Topics

"Create Node Groups From the Actions Menu " on page 312

"In a CSV File, Define Node Groups" on page 316

Specify Node Group Additional Filters

Use the Additional Filters Editor to create expressions that refine the requirements for membership in a Node Group. Make sure to design any complex Additional Filters offline as a Boolean expression first. This method can help to minimize errors when entering your expressions using the Additional Filters Editor.

If any Additional Filters are created, NNMi combines any Device Filters and Additional Filters using the AND Boolean operator as follows:

- NNMi first evaluates any Device Filters. Nodes must match *at least one* Device Filter specification to belong to this Node Group.
- NNMi then evaluates the Additional Filters expression. Nodes *must also match all* Additional Filters expression specifications to belong to this Node Group.

To create an Additional Filters expression:

1. Navigate to the **Node Group Form: Additional Filters** tab.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Expand the **Object Groups** folder.
   c. Select **Node Group**.

   d. Do one of the following:

      - To create a Node Group definition, click the ✳ New icon.

      - To edit a Node Group definition, click the Open icon in the row representing the Node Group definition you want to edit.

   e. In the Node Group form, select the **Additional Filters** tab.

2. Establish the appropriate settings for the Additional Filters you need (see the Additional Filters Editor Components and Additional Filters Editor Buttons table). See "Guidelines for Creating Additional Filters for Node Groups" on page 307 for more information.

   a. Plan out the logic needed for your Filter String.

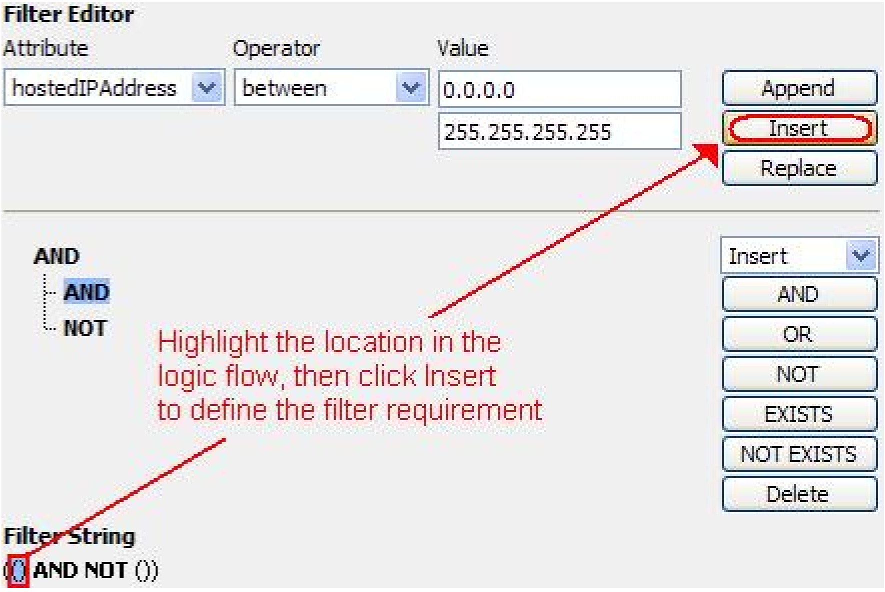
   b. Use the buttons on the bottom half of the Additional Filters Editor to establish the logic structure. See "Add Boolean Operators in the Additional Filters Editor" on page 310.

      For example, to establish the following structure, select **Insert**, then click **AND**, then **NOT**, and then **AND** a second time:

      ```
      (( ) AND NOT ( ))
      ```

   c. Now place your cursor in a location within the displayed Filter String, and use the top half of the filter editor to define the parameters of the selected filter requirement.

      For example, select a set of parentheses and use the Insert button to specify the filter requirement within those parentheses:



3. Click **Save and Close**.

Additional Filters Editor Components for Node Groups

Attribute	Description
Attribute	NNMi provides Additional Filters codes for a subset of object attributes. For more information about the available Additional Filter codes for each NNMi object type, click the link:

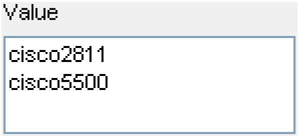
Additional Filters Editor Components for Node Groups, continued

Attribute	Description
	• Node attribute codes [click here for a list of attribute codes] **Values from the Basic Attributes listed on the Node Form**: ▪ hostname (Hostname, *case-sensitive*) ▪ mgmtIPAddress (Management Address) ▪ isSnmpNode (Agent Enabled) ▪ isNnmSystemLocal (NNMi Management Server) ▪ securityGroupName (Security Group) **Note**: If you enter the Name value for a Security Group that you do not have permission to access, the Node Group will be empty. See "Configuring Security" on page 503 for more information. **Values from the Node Form:General Tab**: ▪ sysName (System Name) ▪ sysLocation (System Location) ▪ sysContact (System Contact) ▪ sysOidNode (System Object ID) **Addresses from the Node Form: IP Addresses Tab**: ▪ hostedIPAddress (Address) See "Node Groups of IPv4 or IPv6 Addresses " on page 306 for ideas. **Unique Keys from the Node Form: Capabilities Tab**: ▪ capability (Unique Key of the Capability) **Values from the Node Form: Custom Attributes Tab**: **Note**: When using `customAttrName`and `customAttrValue` pairs, use EXISTS if you want NNMi to consider Nodes that *do not have Custom Attributes* when evaluating the entire Filter String. Otherwise Nodes that do not have Custom Attributes are automatically excluded from the Node Group even if they have values that pass other aspects of your filter. ▪ customAttrName (Custom Attribute Name) ▪ customAttrValue (Custom Attribute Value) • Security Group attribute codes [click here for a list of attribute codes] **Values from the Security Group Form**: **Note**: If you enter the Name or UUID value for a Security Group that you do not have permission to access, the Node Group will be empty. See "Configuring Security" on page 503 for more information. ▪ securityGroupName (Name)

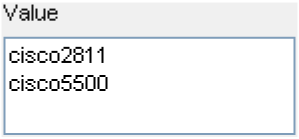
Additional Filters Editor Components for Node Groups, continued

Attribute	Description
	▪ securityGroupUuid (UUID) • Tenant attribute codes [click here for a list of attribute codes] **Values from the Tenant Form**: ▪ tenantName (Name) ▪ tenantUuid (UUID) • Device Profile attribute codes [click here for a list of attribute codes] **Values from the Basics Attributes on the Device Profile Form**: NNMi matches the Label attribute values from the Device Profile Form for each of the following: ▪ devCategoryNode (Device Category) ▪ devVendorNode (Device Vendor) ▪ devFamillyNode (Device Family) To filter on the SNMP system object ID number assigned to a particular make/model, use the sysOidNode attribute. See Values from the Node Form: General Tab. • Regional Manager attribute codes (*NNMi Advanced*) [click here for a list of attribute codes] **Values from the associated entry on the Regional Manager Form: Connection Tab**: ▪ nnmSystemName (Hostname, *case-sensitive*) (*NNMi Advanced*) If the Global Network Management feature is enabled, this attribute value identifies a Regional Manager (NNMi management server).
Operator	The standard query language (SQL) operations to be used for the search. **Note**: Only the `is null` Operator returns null values in its search. Valid operators are described below. • **=** Finds all values equal to the value specified. Click here for an example. Example: `sysName = cisco2811` finds all devices with system name equal to **cisco2811**. • **!=** Finds all values not equal to the value specified. Click here for an example. Example: `sysName != cisco2811` finds all system names other than **cisco2811**. • **<** Finds all values less than the value specified. Click here for an example. IPv4 example: `mgmtIPAddress < 15.239.255.255` finds all IP address values less than **15.239.255.255**

Additional Filters Editor Components for Node Groups, continued

Attribute	Description
	IPv6 example: `mgmtIPAddress < ::ffff:0:0` finds all IP address values less than **::ffff:0:0** • **<=** Finds all values less than or equal to the value specified. Click here for an example. Example: `mgmtIPAddress <= 15.239.255.255` finds all IP address values less than or equal to **15.239.255.255**. • **>** Finds all values greater than the value specified. Click here for an example. IPv4 example: `mgmtIPAddress > 15.238.0.0` finds all IP address values greater than **15.238.0.0** IPv6 example: `mgmtIPAddress > ::ffff:ffff:ffff` finds all IP address values greater than **::ffff:ffff:ffff** • **>=** Finds all values greater than or equal to the value specified. Click here for an example. Example: `mgmtIPAddress >= 15.238.0.0` finds all IP address values greater than or equal to **15.238.0.0**. • **between** Finds all values equal to and between the two values specified. Click here for an example. Example: `mgmtIPAddress between 15.238.0.10 15.238.0.120` finds all IPv4 address values equal to or greater than **15.238.0.10** and equal to or less than **15.238.0.120**. See "Node Groups of IPv4 or IPv6 Addresses " on page 306 for more examples of using the **between** Operator. • **in** Finds any match to at least one value in a list of values. Click here for an example. Example: `sysName in`  Value cisco2811 cisco5500 finds all systems with names that are **cisco2811** or **cisco5500**. **Note**: As shown in the example, each value must be entered on a separate line. NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example (**cisco2811, cisco550**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line. • **is not null** Finds all non-blank values. Click here for an example.

Additional Filters Editor Components for Node Groups, continued

Attribute	Description
	Example: `sysName is not null` finds all systems that have a name value. • **is null** Finds all blank values. Click here for an example. Example: `sysName is null` finds all systems that do not have an assigned name value. • **like** Finds matches using wildcard characters. Click here for more information about using wildcard characters. The following attributes cannot be used with the `like` operator: ▪ hostedIPaddress ▪ mgmtIPaddress The asterisk (*) character means *any number of characters of any type at this location*. **Note**: For optimum performance, avoid beginning your search string with an asterisk (*). The question mark (?) character means *any single character of any type at this location*. Examples: ▪ `sysName like cisco*` finds all system names that begin with **cisco**. ▪ `sysName like cisco??*` finds all system names that *start with* cisco **followed by two characters**. ▪ `sysName like rtr??bld5*` finds all system names that have *specific characters at an exact location*, positions 1-3 (rtr) and 6-9 (bld5). • **not between** finds all values except those between the two values specified. Click here for an example. Example: `mgmtIPAddress not between 15.238.0.10 15.238.0.120` finds all IP address values less than **15.238.0.10** and greater than **15.238.0.120**. See "Node Groups of IPv4 or IPv6 Addresses " on page 306 for more examples of using the **not between** Operator. • **not in** Finds all values except those included in the list of values. Click here for an example. Example: `sysName not in`  Value cisco2811 cisco5500 finds all system name values other than **cisco2811** and **cisco5500**.

Additional Filters Editor Components for Node Groups, continued

Attribute	Description
	Note: As shown in the example, each value must be entered on a separate line. NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example, (**cisco2811, cisco550**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line. • **not like** Finds all that do not have the values specified (using wildcard strings). Click here for an example. The following attributes cannot be used with the `not like` operator: ▪ hostedIPaddress ▪ mgmtIPaddress The asterisk (*) character means *any number of characters of any type at this location*. The question mark (?) character means *any single character of any type at this location*. Examples: ▪ `sysName not like cisco*` finds all system names that do not begin with **cisco**. ▪ `sysName not like cisco??*` finds all system names that do not *begin with* cisco **followed by two characters**. ▪ `sysName not like rtr??bld5*` finds all system names that do not have *specific characters at an exact location*, positions 1-3 (rtr) and 6-9 (bld5).
Value	The value for which you want NNMi to search. Note the following: • The values you enter are case sensitive. • NNMi displays a variable number of value fields depending on the Operator selected. For example, the `between` Operator causes two value fields to be displayed. • The `in` and `not in` operators require that each value be entered on a separate line. • When entering a value for the Capability attribute, copy and paste the Unique Key value from the Node form: Capability tab.

Additional Filters Editor Buttons

Button	Description
Append	Appends the current expression (Attribute, Operator,and Value) to the selected expression already included in the Filter String.

Additional Filters Editor Buttons, continued

Button	Description
Insert	Inserts the current expression (Attribute, Operator,and Value) in front of the cursor location within the Filter String.
Replace	Replaces the selected expression with the expression displayed in the Attribute, Operator, and Value fields.
AND	Inserts the AND Boolean Operator in the selected cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
OR	Inserts the OR Boolean Operator in the current cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
NOT	Can be used in any part of the Filter String to specify that NNMi should exclude nodes with values that pass the expression that immediately follows the NOT. For example, when evaluating the following Filter String, NNMi includes nodes with a hostname that contains **router**, followed by any number of characters, followed by **hp.com** and excludes any nodes with a Device Profile that includes **Cisco** as the Vendor value: `(hostname like router*.hp.com OR NOT (devVendorNode = Cisco))`
EXISTS	Used for filters that include Capabilities or Custom Attribute names and values in the Filer String. Indicates that you want NNMi to consider nodes that do not have any Capabilities or Custom Attributes when evaluating the Filter String. **Note**: If you include Capabilities or Custom Attribute names and values in the Filer String, but do not use EXISTS or NOT EXISTS, NNMi excludes from its search nodes that do not include Capabilities or Custom Attributes. For example, when evaluating the following Filter String, NNMi includes nodes with a hostname that includes **router**, followed by any number of characters, followed by **hp.com** as well as any nodes that have the Custom Attribute **edge** and that edge value is **true**: `(hostname like router*.hp.com OR EXISTS((customAttrName=edge AND customAttrValue=true)))`
NOT EXISTS	Used for filters that include Capabilities or Custom Attribute names and values in the Filer String. Indicates that you want NNMi to consider nodes that do not have any Capabilities or Custom Attributes when evaluating the Filter String, but exclude the nodes that match the expression that follows the NOT EXISTS. **Note**: If you include Capabilities or Custom Attribute names and values in the Filer String, but do not use EXISTS or NOT EXISTS, NNMi excludes from its search nodes that do not include Capabilities or Custom Attributes. For example, when evaluating the following Filter String, NNMi includes nodes with a

Additional Filters Editor Buttons, continued

Button	Description
	hostname that includes **router**, followed by any number of characters, followed by **hp.com** and excludes any nodes with Custom Attribute **edge** and that edge value is **true**. `(hostname like router*.hp.com OR NOT EXISTS ((customAttrName=edge AND customAttrValue=true)))`
Delete	Deletes the selected expression. **Note**: If the Boolean Operator is selected, the Additional Filters Editor deletes all expressions associated with the Boolean Operator.

Node Groups of IPv4 or IPv6 Addresses

Use the Node Group form's Additional Filters editor to create Node Groups based on the following criteria ("Specify Node Group Additional Filters" on page 298):

- All nodes that have *only* IPv4 addresses
  [click here for details of this filter.]

  Both of the following example Node Group's Additional Filters provide the same results. The first example uses IPv4 address notation. The second example uses IPv6 address notation:

  ```
  ((hostedIPAddress between 0.0.0.0 AND 255.255.255.255) AND NOT
  (hostedIPAddress not between 0.0.0.0 AND 255.255.255.255))
  ```

  or (*NNMi Advanced with IPv6 enabled*)

  ```
  ((hostedIPAddress between 0.0.0.0 AND 255.255.255.255) AND NOT
  (hostedIPAddress not between ::ffff:0:0 AND ::ffff:ffff:ffff))
  ```

- All nodes that have *any* IPv4 addresses (could also have IPv6)
  [click here for details of this filter.]

  The following example Node Group's Additional Filter finds any node that has at least one IPv4 address:

  ```
  (hostedIPAddress between 0.0.0.0 AND 255.255.255.255)
  ```

- (*NNMi Advanced with IPv6 enabled*) All nodes that have *only* IPv6 addresses
  [click here for details of this filter.]

  IPv6 addresses extend the number of possible IP addresses. The old IPv4 address range falls within the new IPv6 range. Valid IPv6 address values can be less than or greater than the old IPv4 range of addresses. NNMi Advanced converts the IPv4 addresses to the new IPv6 notation, then stores and filters the IPv4 addresses as IPv6 addresses (`::ffff:a.b.c.d`).

  Both of the following example Node Group's Additional Filters provide the same results. The first example uses IPv4 address notation. The second example uses IPv6 address notation:

```
((hostedIPAddress not between 0.0.0.0 AND 255.255.255.255) AND NOT
(hostedIPAddress between 0.0.0.0 AND 255.255.255.255))
```

or

```
((hostedIPAddress not between ::ffff:0:0 AND ::ffff:ffff:ffff) AND
NOT (hostedIPAddress between 0.0.0.0 AND 255.255.255.255))
```

- (*NNMi Advanced with IPv6 enabled*) All nodes that have *any* IPv6 addresses (could also have IPv4)
  [click here for details of this filter.]

  The following example Node Group's Additional Filter finds any node that has at least one IPv6 address:

  ```
  ((hostedIPAddress between ::0 AND ::fffe:ffff:ffff) OR
  (hostedIPAddress ::1:0:0:0 AND
  ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff))
  ```

- (*NNMi Advanced with IPv6 enabled*) All nodes that have *both* IPv4 and IPv6 addresses (also known as dual-stack nodes)
  [click here for details of this filter.]

  The following example Node Group's Additional Filter finds any node that has at least one IPv4 address and at least one IPv6 address:

  ```
  ((hostedIPAddress between 0.0.0.0 AND 255.255.255.255) AND
  (hostedIPAddress not between 0.0.0.0 AND 255.255.255.255))
  ```

Note: To maximize the performance of Additional Filters based on an IP Address range, avoid multiple filter expressions. For example, use the `between` operator instead of the greater than or equal to (>=) and less than or equal to (<=) operators that cause NNMi to use multiple queries for finding all addresses that match the filter.

Guidelines for Creating Additional Filters for Node Groups

The Additional Filters Editor enables you to create expressions to further define the nodes to be included in a Node Group. Make sure to design any complex Additional Filters offline as a Boolean expression first. This method can help to minimize errors when entering your expressions using the Additional Filters Editor.

When creating Additional Filters for a Node Group, note the following:

- NNMi treats each set of expressions associated with a Boolean Operator as if it were enclosed in parentheses and evaluated together rather than in order of grouping as the nesting implies. Therefore, when using the `AND` operator to combine expressions that include Custom Attributes, include only one customAttrName/customAttrValue pair in the expression. Otherwise, if you use multiple customAttrName and customAttrValue pairs with the `AND` operator, the results might not be as expected. Click here for an example.

In the following example, because the `AND` Boolean operator indicates that NNMi should evaluate all of the `customAttrname` and `customAttrvalue` pairs together, it is not possible for any nodes to match this Additional Filters expression:

Additional Filter Expression Example 1:

```
((customAttrName = capability) AND (customAttrValue =
com.hp.nnm.capability.card.fru)) AND ((customAttrName = location)
AND (customAttrValue = datacenter1))
```

This is because `customAttrName` would need to match both `capability` *and*>`location` at the same time. However, if you use the `OR` operator to combine the `customAttrName` and `customAttrValue` pairs as shown in the following example, the filter should work as expected.

Additional Filter Expression Example 2:

```
((customAttrName = capability) AND (customAttrValue =
com.hp.nnm.capability.card.fru)) OR ((customAttrName = location)
AND (customAttrValue = datacenter1))
```

Using the Node values listed in the following table, all three nodes (nodeA, nodeB, and nodeC) pass the filter in Example 2 because each of these nodes has either the value `com.hp.nnm.capability.card.fru` for `capability`*or* the value `datacenter1` for `location`.

Example Data

Node Name	capabilty	customAttrName	customAttrValuee
nodeA	com.hp.nnm.capability.card.fru	location	datacenter1
nodeB	com.hp.nnm.capability.card.fru	<undefined>	<undefined>
nodeC	<undefined>	location	datacenter1

- Use the EXISTS and NOT EXISTS operators when you want NNMi to consider nodes that do not have any Capabilities or Custom Attributes when evaluating the Filter String. See "Specify Node Group Additional Filters" on page 298 for more information.
- View the expression displayed under **Filter String** to see the logic of the expression as it is created.
- The AND and OR Boolean Operators must contain at least two expressions as shown in the example below.

```
AND
  sysName like cisco*
  sysName != cisco2811
  OR
     sysLocation = Boston
     sysContact In (Johnson,Hickman)
```

NNMi evaluates the expression above as follows:

```
sysName like cisco* AND sysName != cisco2811 AND (sysLocation =
Boston OR sysContact in (Johnson, Hickman))
```

- NNMi finds all nodes with a (system name) sysName beginning with **cisco**, except not **cisco2811**.
- Of these nodes, NNMi then finds all nodes with a (system location) sysLocation of **Boston** or (system contact name) sysContact of **Johnson** or **Hickman**.

- NNMi evaluates only those nodes that contain values for *all* of the attributes included in the Additional Filter expression. Click here for an example.

  If your Node Group filter expression includes the `capability` and `customAttrName` attributes, then NNMi evaluates only nodes that have a value defined for *both* `capability` and `customAttrName`. For example, if you create a Node Group using the following Additional Filters expression, then NNMi evaluates only those nodes that have a value defined for `capability` and a value defined for `customAttrName`:

  ```
  (capability = com.hp.nnm.capability.card.fru) OR (customAttrName =
  location)
  ```

  Using the Node values listed in the following table, NNMi only evaluates nodeA. This is because nodeA contains a value for `capability` and a value for `customAttrName`. NNMi does not evaluate nodeB because it does not have a value for `customAttrName`. NNMi does not evaluate nodeC because it does not have a value for `capability`. NodeA also passes Node Group Additional Filter because its `capability` value of `com.hp.nnm.capability.card.fru` matches the value specified in the Additional Filter expression. Therefore, only nodeA is included in this example Node Group.

  Example Data

Node Name	capabilty	customAttrName	customAttrValuee
nodeA	com.hp.nnm.capability.card.fru	location	datacenter1
nodeB	com.hp.nnm.capability.card.fru	<undefined>	<undefined>
nodeC	<undefined>	location	datacenter1

  Tip: You can populate a placeholder value, such as "none" or "undefined" for any attribute that you want to use in an Additional Filter.

- The placement of your cursor and the subsequent text that is selected is important when performing operations using the Additional Filters Editor. For example, you append to or replace, the expression that is selected.
- The placement of your cursor and the subsequent text that is selected is especially important when adding your Boolean operators. See "Add Boolean Operators in the Additional Filters Editor" on the next page for more information.
- You can drag any of the following items to a new location in the Filter String:
  - Filter Editor Options: AND, OR, NOT, EXISTS, NOT EXISTS
  - Filter Expression (Attribute, Operator and Value)
- When moving items in the Filter String, note the following:

- Click the item you want to move before dragging it to a new location.
- As you drag a selected item, an underline indicates the target location.
- If you are moving the selection up, NNMi places the item above the target location.
- If you are moving the selection down, NNMi places the item below the target location.
- If you attempt to move the selection to an invalid target location, NNMi displays an error message.

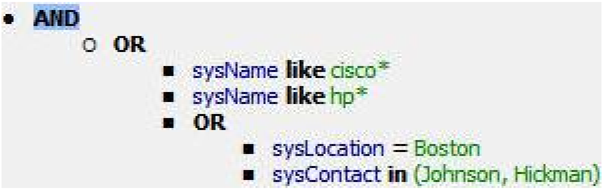
Add Boolean Operators in the Additional Filters Editor

When adding or deleting Boolean Operators using the Additional Filters Editor, note the following:

- Add your highest level Boolean operator first. For example, **AND** is the highest level Boolean operator in the following expression

  (sysName like cisco* OR sysName like hp*) **AND** (sysLocation = Boston OR sysContact in Johnson,Hickman)
- Add each additional Boolean Operator before the expressions to which it applies.
- Select the appropriate Boolean Operator in the expression before you add the expressions to which the Boolean Operator applies.
- When a Boolean Operator is selected and you click **Delete**, any expressions that are associated with the Boolean Operator are also deleted.

  In the example expression below, If you select **AND** and then click **Delete,** the Additional Filters Editor deletes the entire expression.



- **AND**
  - **OR**
    - sysName **like** cisco*
    - sysName **like** hp*
    - **OR**
      - sysLocation = Boston
      - sysContact **in** (Johnson, Hickman)

Click here for an example for creating Node Group Additional Filters.

Node Group Additional Filters Expression Example

```
((sysName like cisco* OR sysName like hp*) AND (sysLocation = Boston
OR sysContact in (Johnson, Hickman)))
```

To add the expression above, after you are in the Additional Filters Editor, follow these steps:

1. Click **AND**.
2. Click **OR**.
3. Select the **OR** you just added to the expression.
4. In the **Attribute** field select **sysName** from the drop-down list.
5. In the **Operator** field, select **like** from the drop-down list.
6. In the **Value** field, enter **cisco***.

7. Click **Append**.
8. In the **Attribute** field, select **sysName** from the drop-down list.
9. In the **Operator** field, select **like** from the drop-down list.
10. In the **Value** field, enter **hp***.
11. Click **Append**.
12. Select the **AND** that you previously added to the expression.
13. Click **OR**.
14. Select the **OR** you just added to the expression.
15. In the **Attribute** field, select **sysLocation** from the drop-down list.
16. In the **Operator** field, select **=** from the drop-down list.
17. In the **Value** field, enter **Boston**.
18. Click **Append**.
19. In the **Attribute** field, select **sysContact** from the drop-down list.
20. In the **Operator** field, select **in** from the drop-down list.
21. In the **Value** field:
    a. enter **Johnson** and press **<Enter>.**
    b. On the new line, enter **Hickman.**
22. Click **Append**.
23. Click **Save** to save your Additional Filters.
24. Select **Actions** > **Preview Members (Current Group Only)** to view the members of the Node Group that is a result of this filter.

    Tip: To test the effects of your Node Group definition on Child Node Groups, in the Node Group form, select **Save**, then **Actions** > **Node Group Details** > **Show Members (Include Child Groups)**. NNMi displays the members of the current Node Group members as well as the members of each associated Child Node Group. Depending on the complexity of your Node Group hierarchy, NNMi might take some time to complete updating the results. Click 🔄 Refresh to check for the most recent changes to Node Group contents.
25. Click 🔄 Refresh to check for the most recent changes to Node Group contents.

Click here for an example for creating an Interface Group Additional Filters.

Interface Group Additional Filters Expression Example

```
((ifName like ATM* AND ifName != ATMS/O/A) AND (ifSpeed = 10 OR
ifSpeed = 100))
```

To add the expression above, follow these steps:

1. Click **AND**.
2. Click **AND**.
3. Select the **AND** you just added to the expression.
4. In the **Attribute** field select **ifName** from the drop-down list.
5. In the **Operator** field, select **like** from the drop-down list.
6. In the **Value** field, enter **ATM***.
7. Click **Append**.
8. In the **Attribute** field, select **ifName** from the drop-down list.
9. In the **Operator** field, select **!=***not equal to* from the drop-down list.
10. In the **Value** field, enter **ATMS/0/A**.
11. Click **Append**.
12. Select the first **AND** that you added to the expression.
13. Click **OR**.
14. Select the **OR** you just added to the expression.
15. In the **Attribute** field, select **ifSpeed** from the drop-down list.
16. In the **Operator** field, select **=** from the drop-down list.
17. In the **Value** field, enter **10**.
18. Click **Append**.
19. In the **Attribute** field, select **ifSpeed** from the drop-down list.
20. In the **Operator** field, select **=** from the drop-down list.
21. In the **Value** field, enter **100**.
22. Click **Append**.
23. Click **Save** to save your Additional Filters.
24. Select **Actions** >**Show Members** to view the members of the Interface Group that is a result of this filter.

Tip: You can right-click any object in a table or map view to access the **Actions** menu.

Create Node Groups From the Actions Menu

Node Groups are used for a variety of purposes in NNMi. See "Creating Groups of Nodes or Interfaces" on page 294 for more information.

Note: By default NNMi Administrators can create, modify, and delete Node Groups. NNMi Administrators can configure NNMi to permit User Accounts assigned to the NNMi Operator Level 2 User Group to create, modify, and delete Node Groups. See the *HP Network Node*

> *Manager i Software Deployment Reference* for more information (**Help** → **Documentation Library**). Search for "Node Group".

You can create any number of Node Groups in addition to the ones that NNMi provides (see "Node Groups Provided by NNMi" on page 335).

You can easily create a Node Group from any Nodes or map view using the **Actions** menu. NNMi adds the selected nodes to the Node Group that it creates. When creating Node Groups using the **Actions** menu, note the following:

- Multiple nodes can be associated with one Node Group.
- One node can be associated with multiple Node Groups.
- If you change the Node Group name, the Group Membership does not change.

To create a Node Group from the Actions menu (if your role permits you to do this):

1. Navigate to a **Node** inventory view.
   a. From the workspace navigation panel, select the **Inventory** workspace.
   b. Select the node view of interest (for example, **Nodes** view).

   > **Tip:** You can also select Nodes from a map view.

2. Use Ctrl-Click to select each node you want to add to a Node Group.
3. Select **Actions** → **Node Group Membership**.
4. Select **Add to a new Node Group**.
5. In the **Node Group Membership** dialog, box, enter the Name of the Node Group you want to create. This name is a maximum of 255 characters. Alpha-numeric, spaces, and special characters (~ ! @ # $ % ^ & * ( ) _+ -) are permitted.

   > **Tip:** If you want to view the associated Node Group form, in the Node Group Membership dialog box, check ☑ **Open the Node Group form**. When selected, NNMi opens the Node Group form so that you can view your changes.

   > **Note:** NNMi lists the nodes you have added on the **Additional Nodes** tab.

6. In the **Node Group Membership** dialog box, click **OK** to save your changes.

   If you configured this Node Group for Monitoring, NNMi applies your changes during the next monitoring cycle. "Configure NNMi Monitoring Behavior" on page 340.

   If you specified to open the Node Group form and then made additional changes, click 💾**Save and Close** to save your changes.

To review a Node Group definition:

1. From the workspace navigation panel, select the **Inventory** workspace.
2. Select the **Node Groups** view.

3. Double-click the row representing the Node Group definition you want to see.
4. The Node Group form displays.

   > **Note:** NNMi monitors the status of each Node Group over time. To check Node Group status information, access the Node Group form's Status tab.

5. When finished, click the Close icon.

You can also use the nnmloadnodegroups.ovpl command to list the following:

- Names of the existing Node Groups
- Selected attributes of nodes that are members of a specified Node Group

Special Actions are available for Node Groups and Interface Groups.

Related Topics

"Create Node Groups Using Filters or Hostname Lists (Configuration: Node Groups)" on page 296

"In a CSV File, Define Node Groups" on page 316

Add Nodes to a Node Group From the Actions Menu

Node Groups are used for a variety of purposes in NNMi. See "Creating Groups of Nodes or Interfaces" on page 294 for more information.

> **Note:** By default NNMi Administrators can create, modify, and delete Node Groups. NNMi Administrators can configure NNMi to permit User Accounts assigned to the NNMi Operator Level 2 User Group to create, modify, and delete Node Groups. See the *HP Network Node Manager i Software Deployment Reference* for more information (**Help** → **Documentation Library**). Search for "Node Group".

You can create any number of Node Groups in addition to the ones that NNMi provides (see "Node Groups Provided by NNMi" on page 335).

You can easily add one or more Nodes to a Node Group from any Nodes or map view using the **Actions** menu. NNMi adds the selected nodes to the Node Group specified.

When adding Nodes to an existing Node Groups using the **Actions** menu, note the following:

- Multiple nodes can be associated with one Node Group.
- One node can be associated with multiple Node Groups.
- If you change the Node Group name, the Group Membership does not change.

> **Tip:** *NNM iSPI Performance for Metrics only*. NNMi automatically synchronizes Interface Group and Node Group configuration changes between NNMi and NNM iSPI Performance for Metrics. However, in some cases, additional configuration changes that affect Node Group or Interface Group membership might take longer to synchronize. If you do not see one or more

> nodes in an NNMi Performance for Metrics report that are visible in NNMi, use the **Actions** → **HP NNM iSPI Performance** → **Sync Interface and Node Groups** with NNMi option. This option forces NNMi to synchronize the Interface and Node Group information between NNMi and NNM iSPI Performance for Metrics more quickly than the default time frame.

To create a Node Group using the Actions menu (if your role permits you to do this):

1. Navigate to a **Nodes** inventory view.
   a. From the workspace navigation panel, select the **Inventory** workspace.
   b. Select the node view of interest (for example, **Nodes** view).

   > **Tip:** You can also select Nodes from a map view.

2. Use Ctrl-Click to select each node you want to add to a Node Group.
3. Select **Actions** → **Node Group Membership**
4. Select **Add to an existing Node Group**.
5. In the **Node Group Membership** dialog, box, select the Lookup icon and select one of the options from the drop-down menu:

   Quick Find to view and select from the list of all existing Node Groups.

   Open to display the details of a selected Node Group.

   > **Tip:** If you want to view the associated Node Group form, in the Node Group Membership dialog box, check ☑ **Open the Node Group form**.

   > **Note:** NNMi adds the nodes on the **Additional Nodes** tab. NNMi automatically opens the Node Group form so that you can make any additional changes.

6. In the **Node Group Membership** dialog box, click **OK** to save your changes.

   If you configured this Node Group for Monitoring, NNMi applies your changes during the next monitoring cycle. "Configure NNMi Monitoring Behavior" on page 340.

   If you specified to open the Node Group form and then made additional changes, click **Save and Close** to save your changes.

To review a Node Group definition:

1. From the workspace navigation panel, select the **Inventory** workspace.
2. Select the **Node Groups** view.
3. Double-click the row representing the Node Group definition you want to see.
4. The Node Group form displays.

   > **Note:** NNMi monitors the status of each Node Group over time. To check Node Group

> status information, access the Node Group form's Status tab.

5. When finished, click the Close icon.

You can also use the nnmloadnodegroups.ovpl command to list the following:

- Names of the existing Node Groups
- Selected attributes of nodes that are members of a specified Node Group

Special Actions are available for Node Groups and Interface Groups.

Related Topics

"Create Node Groups Using Filters or Hostname Lists (Configuration: Node Groups)" on page 296

"In a CSV File, Define Node Groups" below

In a CSV File, Define Node Groups

Node Groups are used for a variety of purposes in NNMi. See "Creating Groups of Nodes or Interfaces" on page 294 for more information.

> **Note:** By default NNMi Administrators can create, modify, and delete Node Groups. NNMi Administrators can configure NNMi to permit User Accounts assigned to the NNMi Operator Level 2 User Group to create, modify, and delete Node Groups. See the *HP Network Node Manager i Software Deployment Reference* for more information (**Help** → **Documentation Library**). Search for "Node Group".

You can create any number of Node Groups in addition to the ones that NNMi provides (see "Node Groups Provided by NNMi" on page 335).

You can create a Node Group by either using the NNMi console or a comma separated values (CSV) file. For example, if you have Node Group information in a Microsoft Excel spreadsheet, you can save this information as a .csv file and use the `nnmloadnodegroups.ovpl` command to add this node group information to NNMi.

> **Tip:** See the nnmloadnodegroups.ovpl Reference Page for more information about the `nnmloadnodegroups.ovpl` command, including requirements for the CSV file. You must provide a CSV file with a specific syntax and order. Each column in the CSV file has a pre-defined meaning as described in the nnmloadnodegroups.ovpl Reference Page.

To create a Node Group using a comma separated values (CSV) text file, use the `nnmloadnodegroups.ovpl` command:

> **Tip:** If your goal is to *merge* new information into an existing Node Group, use nnmloadnodegroups.ovpl to create a *new Node Group* with the additional settings. Then use the Node Group form to assign that new Node Group as a *Child Node Group* of the original Node Group.

```
nnmloadnodegroups.ovpl -r [true|false] -u <NNMiadminUsername> -p
<NNMiadminPassword> -f <CSV file name>
```

CSV file name is the name of the CSV file that contains the Node Group information.

`-r true` means *all the settings* for any existing Node Group with the same `Name` are overwritten with the values in your CSV file.

> **Note:** This is not a merge. It is a complete replacement of that Node Group configuration.

`-r false` (defautl) means if the Node Group `Name` already exists, the `nnmloadnodegroups.ovpl` command does not change the previous settings.

To create Interface Groups using a CSV file, see "In a CSV File, Define Interface Groups" on page 334

Remove Nodes from Node Groups

NNMi enables you to remove one or more nodes from a selected Node Group or from all of the Node Groups to which they belong.

To remove one or more nodes from an Node Group (if your role permits you to do this):

1. Navigate to an **Nodes** inventory view.
   a. From the workspace navigation panel, select the **Inventory** workspace.
   b. Select the node view of interest (for example, **Nodes** view).

      Tip: You can also select Nodes from a map view.
2. Use CTRL-Click to select each node you want to remove from a Node Group.
3. Select **Actions** → **Node Group Membership** → **Remove from a Node Group...**.
4. In the **Node Group Membership** dialog, box, select the Lookup icon, and then Quick Find to select the Node Group from which you want to remove the selected nodes.
5. In the **Node Group Membership** dialog box, click **OK**.

> **Tip:** *NNM iSPI Performance for Metrics only*. NNMi automatically synchronizes Interface Group and Node Group configuration changes between NNMi and NNM iSPI Performance for Metrics. However, in some cases, additional configuration changes that affect Node Group or Interface Group membership might take longer to synchronize. If you do not see one or more nodes in an NNMi Performance for Metrics report that are visible in NNMi, use the **Actions** → **HP NNM iSPI Performance** → **Sync Interface and Node Groups** with NNMi option. This option forces NNMi to synchronize the Interface and Node Group information between NNMi and NNM iSPI Performance for Metrics more quickly than the default time frame.

Related Topics

"Create Node Groups From the Actions Menu " on page 312

Configure Node Group Status

NNMi enables an NNMi administrator to configure the Node Group status calculations using either of the following methods:

- Assign the Node Group the most severe status of any Node Group member. This is the default method for obtaining Node Group Status.
- Configure the percentage thresholds for one or more Node Group target statuses. For example, when defining percentage values for a target status of **Critical**, you might change the default so that 30 percent of the nodes in the group must have a status other than Normal, for the Node Group Status to be **Critical**.

> **Tip:** Use the **Actions** → **Status Details** to see how NNMi calculates the status for a selected Node Group.

> **Tip:** You can right-click any object in a table or map view to access the **Actions** menu.

To configure Node Group status calculations, do the following:

1. Navigate to the **Status Configuration** form.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Select **Status Configuration.**
2. Make one of the following configuration choices:
   - To assign the Node Group the most severe Status of any Node Group member, in the **Status Configuration** form, under **Global Control,** make sure **Propagate Most Severe Status** is checked:

     Propagate Most Severe Status ☑
   - To configure percentage values for a Node Group Target Status, do the following:
     i. In the **Status Configuration** form, under **Global Control,** make sure the **Propagate Most Severe Status** is cleared:

        Propagate Most Severe Status ☐
     ii. Configure the percentage values for a Node Group Target Status
3. Click **Save and Close**.

   NNMi applies your changes after the configuration is saved. Node Group status is updated anytime a Node Group membership changes.

Configure Percentage Values for the Target Status

NNMi enables you to configure how the status of a Node Group is calculated.

> **Note:** The percentage is calculated using only those nodes in the Node Group that have a

Management Mode value of **Managed**. For example, if a Node Group includes 10 nodes and 3 of the nodes are **Not Managed**, 5 of the nodes have a Status of **Normal**, and 2 have a status of **Critical,** the percentage of **Critical** nodes is 2/7 * 100.

To configure the percentage values for a Node Group Target Status:

1. Navigate to the **Status Configuration** form.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Select **Status Configuration.**
2. Locate the **Node Group Status Settings** tab.
3. Do one of the following:
   - To create a Node Group Status Settings definition, click the ✱ New icon.
   - To edit a Node Group Status Settings definition, select a row and click the 📂 Open icon.
   - To delete a Node Group Status Settings definition, select a row and click the ✖ Delete button
4. Establish the appropriate settings to identify this Node Group Status Settings definition. (See the **"Node Group Status Settings Form" below**.)

Note: You can only define one configuration for each Target Status.

Node Group Status Settings Form

The Node Group Status Settings form is used to configure the percentage thresholds for a Node Group Target Status. The percentage thresholds you specify define what percentage of nodes within the group must have a particular Status. When the percentage thresholds are reached, the Node Group is assigned the associated Target Status. For example, when defining percentage thresholds for a target status of **Critical**, you might change the default so that 10 percent of the nodes in the group must have a status of **Critical** for the Node Group Status to be **Critical**.

Note: Use a percentage threshold between 0 (zero) and 1 (for example, .01) to indicate the Target Status to be reached when one node in the Node Group reaches a specified Status. For example, if you want the Node Group Status to be set to **Critical** when the Status of one node in the Group becomes **Critical**, enter a percentage less than one for the **Critical %** value.

To define percentage thresholds for a Target Status:

1. Navigate to the **Node Group Status Settings** form.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Select **Status Configuration**.
   c. Navigate to the **Node Group Status Settings** tab.

d. Do one of the following:

   - To create a Node Group Status Settings definition, click the ✳ New icon.
   - To edit a Node Group Status Settings definition, select a row and click the Open icon.
   - To delete a Node Group Status Settings definition, select a row and click the ✖ Delete icon.

2. Set the Target Status and percentages you want (see Basic Attributes table).
3. Click **Save and Close** to return to the **Status Configuration** form.
4. Click **Save and Close**. NNMi applies your changes after the configuration is saved.

Basics Attributes

Attribute	Description
Target Status	The Status you are configuring. This Status is assigned to the Node Group whenever the specified percentage thresholds are reached. Note the following: • Whether all or one of the percentage thresholds must be reached for a Target Status configuration depends on the Boolean operator you select. The default Boolean operator is OR. (Also see Combine with AND below.) • If you do not specify any percentages for a Target Status, it does not appear as a Status for a Node Group.
Critical %	Specifies the required percentage of nodes in the group that must have a Status value set to **Critical** before NNMi assigns the Target Status.
Major %	Specifies the required percentage of nodes in the group that must have a Status value set to **Major** before NNMi assigns the Target Status.
Minor %	Specifies the required percentage of nodes in the group that must have a Status value set to **Minor** before NNMi assigns the Target Status.
Warning %	Specifies the required percentage of nodes in the group that must have a Status value set to **Warning** before NNMi assigns the Target Status.
Non-Normal %	Specifies the required percentage of nodes in the group that must have a Status value set to any of the following before NNMi assigns the Target Status: • Critical • Major • Minor • Warning
Unknown %	Specifies the required percentage of nodes in the group that must have a Status value set to **Unknown** before NNMi assigns the Target Status.
Combine	Specifies that you want NNMi to combine the percentage thresholds you enter using

Basics Attributes, continued

Attribute	Description
with AND	the AND Boolean operator. When using this option, note the following: • All percentage thresholds you enter must be reached for the Node Group to be assigned the Target Status. • The percentage thresholds you enter must not exceed 100 percent.

Create Interface Groups

Interface Groups are used for a variety of purposes in NNMi. See "Creating Groups of Nodes or Interfaces" on page 294 for more information.

You can create any number of Interface Groups in addition to the ones that NNMi provides (see "Interface Groups Provided by NNMi" on page 338).

> **Tip:** *NNM iSPI Performance for Metrics only*. NNMi automatically synchronizes Interface Group and Node Group configuration changes between NNMi and NNM iSPI Performance for Metrics. However, in some cases, additional configuration changes that affect Node Group or Interface Group membership might take longer to synchronize. If you do not see one or more nodes in an NNMi Performance for Metrics report that are visible in NNMi, use the **Actions** → **HP NNM iSPI Performance** → **Sync Interface and Node Groups** with NNMi option. This option forces NNMi to synchronize the Interface and Node Group information between NNMi and NNM iSPI Performance for Metrics more quickly than the default time frame.

To create Interface Groups, use one or more of the following methods:

- "Create Interface Groups Using ifType Values and Filters (Configuration: Interface Groups)" below
- "In a CSV File, Define Interface Groups" on page 334

Related Topics

"Troubleshooting Interface Changes" on page 335

"Create Node Groups" on page 295

Create Interface Groups Using ifType Values and Filters (Configuration: Interface Groups)

Interface Groups are used for a variety of purposes in NNMi. See "Creating Groups of Nodes or Interfaces" on page 294 for more information.

You can create any number of Interface Groups in addition to the ones that NNMi provides (see "Interface Groups Provided by NNMi" on page 338).

One method for creating Interface Group is using ifType values or Filters to match the way your team identifies important network devices. For example, each interface group can include one or more interface-type specifications (based on industry-standard IANA ifType-MIB variables).

When determining membership in this Interface Group, NNMi combines the results of all Interface Group configuration settings in the following manner:

- NNMi first evaluates ifType Filters. If any exist, interfaces must match *at least one* specification to belong to this Interface Group.
- NNMi then evaluates any Additional Filters. Interfaces *must also pass all* Additional Filters specifications to belong to this Interface Group.
- If a Node Group is specified for this Interface Group, any interface in this group must be contained in a node that is a member of the Node Group specified in the Basics section.

To define an Interface Group using ifTypevalues or Filters (if your role permits you to do this):

1. Navigate to the **Interface Group** form.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Expand the **Object Groups** folder.
   c. Select the **Interface Groups** view.
2. Do one of the following:
   - To create an Interface Group, click the New icon.
   - To edit an Interface Group, click the Open icon in the row representing the Interface Group you want to edit.
3. Provide the Basics for this interface group, such as Name, Notes, and behavior designations (see Interface Group Form help).
4. *Optional.* Navigate to the **ifType Filters** tab.

   Identify one or more interface types that belong to this group:
   - To add an `ifType` filter, click the New icon, and continue.
   - To change an `ifType` filter, click the Open icon in the row representing the configuration you want to edit, and continue.
   - To delete an `ifType`filter, select a row and click the Delete icon.
5. In the ifType Filter form, click the Lookup icon and select one of the options from the drop-down menu:
   - Show Analysis to view Analysis Pane information for the currently selected `ifType` . (See Use the Analysis Pane for more information about the Analysis Pane.)
   - Quick Find to view and select from the list of all existing `ifType` values (for more information see "Use the Quick Find Window" on page 41).
   - Open to display the details of the currently selected `ifType`.

- ✱ New to create a new `ifType` (see "Add New ifType Values (Interface Types) to the List" on page 333).

6. *Optional.* Navigate to the **Additional Type Filters** tab.

   Use the Additional Filters Editor to filter based on the current values of a subset of Interface object attributes. See "Specify Interface Group Additional Filters" below.

7. Click **Save and Close** to return to the Interface Group form.

   Note: You must click **Save and Close** to save your changes each time you create an Interface Group.

8. Click **Save and Close**.

   If you configured this Interface Group for Monitoring, NNMi applies your changes during the next monitoring cycle. See "Configure NNMi Monitoring Behavior" on page 340.

To review an Interface Group definition:

1. From the workspace navigation panel, select the **Inventory** workspace.
2. Select the **Interface Groups** view.
3. Double-click the row representing the Interface Group.
4. The Interface Group form displays.
5. When finished, click the Close icon.

Special Actions are available for Node Groups and Interface Groups.

Specify Interface Group Additional Filters

The Additional Filters Editor enables you to create expressions to further define the interfaces to be included in an Interface Group. Make sure to design any complex Additional Filters offline as a Boolean expression first. This method can help to minimize errors when entering your expressions using the Additional Filters editor.

If any Additional Filters are created:

- NNMi first evaluates any Interface Type filter. Nodes must match *at least one* specification to belong to this Interface Group.
- NNMi then evaluates the Additional Filters expression. Nodes *must also match all* Additional Filters expression specifications to belong to this Interface Group.

To create any Additional Filters expression:

1. Navigate to the **Interface Group Form: Additional Filters** tab.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Expand the **Object Groups** folder.
   c. Select **Interface Groups**.
   d. Do one of the following:

- To create an Interface Group definition, click the ✱ New icon.
- To edit an Interface Group definition, click the Open icon in the row representing the configuration you want to edit.

e. In the Interface Group form, select the **Additional Filters** tab.

2. Establish the appropriate settings for the Additional Filters you need. (See the Additional Filters Editor Components, Additional Filters Editor Buttons table. See also "Guidelines for Creating Additional Filters for Interface Groups" on page 332.)

   a. Plan out the logic needed for your Filter String.

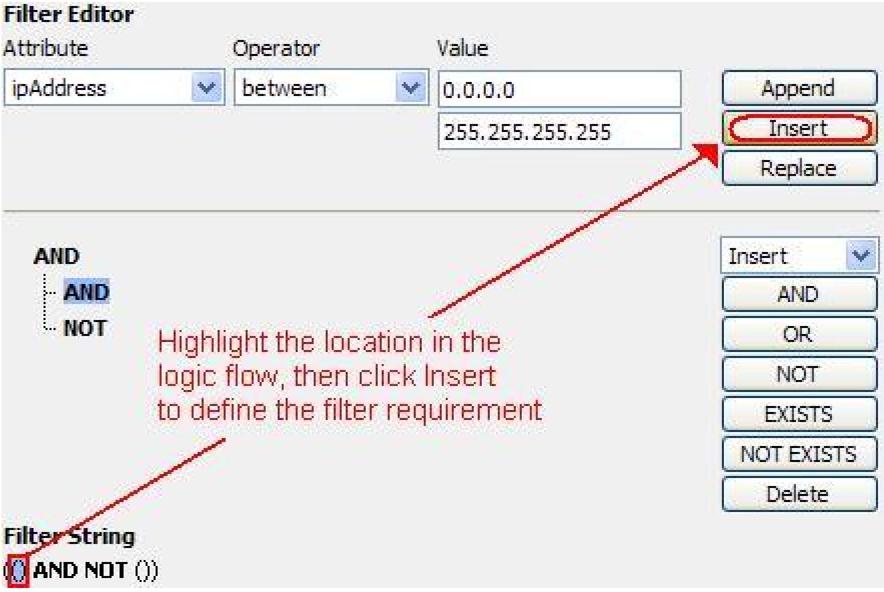
   b. Use the buttons on the bottom half of the Additional Filters Editor to establish the logic structure.

   For example, to establish the following structure, select **Insert**, then click **AND**, then **NOT**, and then **AND** a second time:

   ```
   (( ) AND NOT ( ))
   ```

   c. Now place your cursor in a location within the displayed Filter String, and use the top half of the filter editor to define the parameters of the selected filter requirement.

   For example, select a set of parentheses and use the Insert button to specify the filter requirement within those parentheses:



0.0.0.0

255.255.255.255

(() AND NOT ())

3. Click **Save and Close**.

Additional Filters Editor Components for Interface Groups

Attribute	Description
Attribute	NNMi provides Additional Filters codes for a subset of object attributes. For more information about each one, click the link: • Interface attribute codes [click here for a list of attribute codes] **Values from the Basic Attributes listed on the Interface Form**:

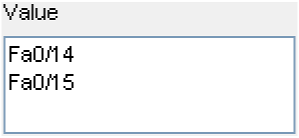
Additional Filters Editor Components for Interface Groups, continued

Attribute	Description
	▪ ifName (Name) ▪ hostedOn (Hosted On Node) ▪ ifPhysAddress (Physical Address) **Values from the Interface Form: General Tab**: ▪ ifAlias (InterfaceAlias) Note the following when using the ifAlias attribute: ○ To include empty (or null) ifAlias entries in your search criteria, match the value "null" (for example: `ifAlias is null`) ○ If you search for an empty ifAlias in your search criteria, the empty value will not be matched (for example do not use : `ifAlias != <`*`string`*`>`) ▪ ifDesc (InterfaceDescription) ▪ ifIndex (InterfaceIndex) ▪ ifSpeed (Interface Speed) **Addresses from the Interface Form: IP Addresses Tab**: ▪ ipAddress (IP Address associated with the interface) See "Interface Groups of IPv4 or IPv6 Addresses" on page 331 for ideas. **Unique Keys from the Interface Form: Capabilities Tab**: ▪ capability (Unique Key of the Capability) **Values from the Interface Form: Custom Attributes Tab**: **Note**: When using `customAttrName`and `customAttrValue` pairs, use EXISTS if you want NNMi to consider Nodes that *do not have Custom Attributes* when evaluating the entire Filter String. Otherwise Nodes that do not have Custom Attributes are automatically excluded from the Node Group even if they have values that pass other aspects of your filter. ▪ customAttrName (Custom Attribute Name) ▪ customAttrValue (Custom Attribute Value) • Node attribute codes [click here for a list of attribute codes] **Values from the Basics information on the Node Form:** ▪ isSnmpInterface (Agent Enabled) **Values from the Node Form: General Tab.** ▪ sysOidInterface (System Object ID) • Device Profile attribute codes [click here for a list of attribute codes] **Values from the Basics information on the Device Profile Form**:

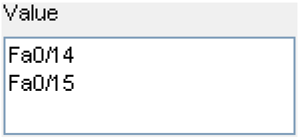
Additional Filters Editor Components for Interface Groups, continued

Attribute	Description
	NNMi matches the Label attribute values from the Device Profile Form for each of the following: ▪ devCategoryInterface (Device Category) ▪ devVendorInterface (Device Vendor) ▪ devFamilyInterface (Device Family) To filter on the parent node's SNMP system object ID number (assigned to a particular make/model), use the sysOidInterface attribute. See Values from the Interface Form: General Tab. • VLAN attribute codes [click here for a list of attribute codes] **Values from the Basic Attributes on the VLAN form:**: **Note**: To maximize performance, when you want to filter interfaces based on a VLAN Id or VLAN Name, avoid using multiple filter expressions. For example, use the `between` operator instead of the greater than or equal to (>=) and less than or equal to (<=) operators. ▪ vlanid (VLAN Id) ▪ vlanName (Global VLAN Name) • Port attribute codes [click here for a list of attribute codes] **Values from the Basic Attributes on the Port form:**: **Note**: If the interface has multiple ports, the interface is selected if there is a match on any one port associated with the interface. ▪ configuredDuplexSetting (Configured Duplex Setting) See Port form for a list of possible values.
Operator	The standard query language (SQL) operations to be used for the search. **Note**: Only the `is null` Operator returns null values in its search. Valid operators are described below. • **=** Finds all values equal to the value specified. Click here for an example. Example: `ifName=Fa0/14` finds all interface names that are equal to **Fa0/14**. • **!=** Finds all values not equal to the value specified. Click here for an example. Example:`ifName != lan0` finds all interface names other than **lan0**. • **<** Finds all values less than the value specified. Click here for an example. Example: `ifSpeed <= 100000000` finds all interfaces with an (interface speed) ifSpeed less than **100 Mbps**. • **<=** Finds all values less than or equal to the value specified. Click here for an example.

Additional Filters Editor Components for Interface Groups, continued

Attribute	Description
	Example: `ifSpeed <= 100000000` finds all interfaces with an (interface speed) ifSpeed less than or equal to **100 Mbps**. - **>** Finds all values greater than the value specified. Click here for an example. Example: `ifSpeed >= 10000000` finds all interfaces with an (interface speed) ifSpeed greater than **10 Mbps**. - **>=** Finds all values greater than or equal to the value specified. Click here for an example. Example: `ifSpeed >= 10000000` finds all interfaces with an (interface speed) ifSpeed greater than or equal to **10 Mbps**. - **between** Finds all values equal to and between the two values specified. Click here for an example. Example: `ifSpeed between 10000000 100000000` finds all interfaces with an (interface speed) ifSpeed equal to or greater than **10 Mbps** and equal to or less than **100 Mbps**. See "Interface Groups of IPv4 or IPv6 Addresses" on page 331 for more examples of using the **between** Operator. - **in** Finds any match to at least one value in a list of values. Click here for an example. Example: `ifName in`  Value Fa0/14 Fa0/15 finds all interfaces with names that are **Fa0/14** or **Fa0/15**. **Note**: As shown in the example, each value must be entered on a separate line. NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example, (**Fa0/14, Fa0/15**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line. - **is not null** Finds all non-blank values. Click here for an example. Example: `ifName is not null` finds all interfaces that have a name value. - **is null** Finds all blank values. Click here for an example. Example:`ifName is null` finds all interfaces that do not have an assigned name value. - **like** Finds matches using wildcard characters. Click here for more information about using wildcard characters. The following attributes cannot be used with the `like` operator:

Additional Filters Editor Components for Interface Groups, continued

Attribute	Description
	■ ifIndex ■ ifSpeed ■ IPAddress The asterisk (*) character means *any number of characters of any type at this location*. The question mark (?) character means *any single character of any type at this location*. Examples: ■ `ifName like ATM*` finds all interface names that begin with **ATM**. ■ `ifName like Ethernet??*` finds all interface names that *begin with***Ethernet** followed by two characters. ■ `ifName like 10/???BASE-TX*` finds all interface names that have *specific characters at an exact location*, positions 1-3 (10/) and 7-13 (BASE-TX). • **not between** Finds all values except those between the two values specified. Click here for an example. Example: `ifSpeed not between 10000000 100000000` finds all interfaces with an (interface speed) ifSpeed less than **10 Mbps** and greater than **100 Mbps**. See "Interface Groups of IPv4 or IPv6 Addresses" on page 331 for more examples of using the **not between** Operator. • **not in** Finds all values except those included in the list of values. Click here for an example. Example: `ifName not in`  Value Fa0/14 Fa0/15 finds all interface name values other than **Fa0/14** or **Fa0/15**. **Note**: As shown in the example, each value must be entered on a separate line. NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example, (**Fa0/14, Fa0/15**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line. • **not like** Finds all that do not have the values specified (using wildcard strings). Click here for an example. The following attributes cannot be used with the `not like` operator:

Additional Filters Editor Components for Interface Groups, continued

Attribute	Description
	▪ ifIndex ▪ ifSpeed ▪ IPAddress The asterisk (*) character means *any number of characters of any type at this location*. The question mark (?) character means *any single character of any type at this location*. Examples: ▪ `ifName not like ATM*` finds all interface names that do not begin with **ATM**. ▪ `ifName not like Ethernet??*` finds all interface names that do not *begin with***Ethernet** followed by two characters. ▪ `ifName not like 10/???BASE-TX*` finds all interface names that do not have *specific characters at an exact location*, positions 1-3 (10/) and 7-13 (BASE-TX).
Value	The value for which you want NNMi to search. Note the following: • The values you enter are case sensitive. • NNMi displays a variable number of value fields depending on the Operator selected. For example, the `between` Operator causes two value fields to be displayed. • The `in` and `not in` operators require that each value be entered on a separate line. • When entering a value for the Capability attribute, copy and paste the Unique Key value from the Interface form: Capability tab.

Additional Filters Editor Buttons

Button	Description
Append	Appends the current expression (Attribute, Operator,and Value) to the selected expression already included in the Filter String.
Insert	Inserts the current expression (Attribute, Operator,and Value) in front of the cursor location within the Filter String.
Replace	Replaces the selected expression with the expression displayed in the Attribute, Operator, and Value fields.
AND	Inserts the AND Boolean Operator in the selected cursor location.

Additional Filters Editor Buttons, continued

Button	Description
	Note: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
OR	Inserts the OR Boolean Operator in the current cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
NOT	Can be used in any part of the Filter String to specify that NNMi should exclude interfaces with values that pass the expression that immediately follows the NOT. For example, when evaluating the following expression, NNMi includes interfaces with (interface description) ifDesc containing **VLAN**, and excludes any Interfaces that have **VLAN10** for the (interface name) ifName value: `(ifDesc like VLAN AND NOT (ifName=VLAN10))` **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
EXISTS	Used for filters that include Capabilities or Custom Attribute names and values in the Filer String. Indicates that you want NNMi to consider interfaces that do not have any Capabilities or Custom Attributes when evaluating the Filter String. **Note**: If you include Capabilities or Custom Attribute names and values in the Filer String, but do not use EXISTS or NOT EXISTS, NNMi excludes from its search interfaces that do not include Capabilities or Custom Attributes. For example, when evaluating the following Filter String, NNMi includes interfaces with (interface description) ifDesc containing **VLAN**, as well as any Interfaces Custom Attribute **Role** value is **LAN Connection to Oracle Server**: `(ifDesc like VLAN OR EXISTS((customAttrName=Role AND customAttrValue=LAN Connection to Oracle Server)))` **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
NOT EXISTS	Used for filters that include Capabilities or Custom Attribute names and values in the Filer String. Indicates that you want NNMi to consider interfaces that do not have any Capabilities or Custom Attributes when evaluating the Filter String, but exclude the interfaces that match the expression that follows the NOT EXISTS. **Note**: If you include Capabilities or Custom Attribute names and values in the Filer String, but do not use EXISTS or NOT EXISTS, NNMi excludes from its search interfaces that do not include Capabilities or Custom Attributes. For example, when evaluating the following expression, NNMi includes interfaces with (interface description) ifDesc containing **VLAN**, and excludes any Interfaces that have the Custom Attribute **Role** and that Role value is **LAN Connection to Oracle Server**: `(ifDesc like VLAN OR NOT EXISTS((customAttrName=Role AND`

Additional Filters Editor Buttons, continued

Button	Description
	`customAttrValue=LAN Connection to Oracle Server)))` **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
Delete	Deletes the selected expression. **Note**: If the Boolean Operator is selected, the Additional Filters Editor deletes all expressions associated with the Boolean Operator.

Interface Groups of IPv4 or IPv6 Addresses

Use the Interface Group form's Additional Filters Editor to create Interface Groups based on the following criteria ("Specify Interface Group Additional Filters" on page 323):

- All interfaces that have *only* IPv4 addresses
  [click here for details of this filter.]

  Both of the following example interface Group's Additional Filters provide the same results. The first example uses IPv4 address notation. The second example uses IPv6 address notation:

  ```
  ((ipAddress between 0.0.0.0 AND 255.255.255.255) AND NOT (ipAddress
  not between 0.0.0.0 AND 255.255.255.255))
  ```

  or (*NNMi Advanced with IPv6 enabled*)

  ```
  ((ipAddress between 0.0.0.0 AND 255.255.255.255) AND NOT (ipAddress
  not between ::ffff:0:0 AND ::ffff:ffff:ffff))
  ```

- All interfaces that have *any* IPv4 addresses (could also have IPv6)
  [click here for details of this filter.]

  The following example interface Group's Additional Filter finds any interface that has at least one IPv4 address:

  ```
  (ipAddress between 0.0.0.0 AND 255.255.255.255)
  ```

- (*NNMi Advanced with IPv6 enabled*) All interfaces that have *only* IPv6 addresses
  [click here for details of this filter.]

  IPv6 addresses extend the number of possible IP addresses. The old IPv4 address range is within the new IPv6 range. Valid IPv6 address values can be less than or greater than the old IPv4 range of addresses. NNMi Advanced converts the IPv4 addresses to the new IPv6 notation, then stores and filters the IPv4 addresses as IPv6 addresses (`::ffff:a.b.c.d`).

  Both of the following example interface Group's Additional Filters provide the same results. The first example uses IPv4 address notation. The second example uses IPv6 address notation:

  ```
  ((ipAddress not between 0.0.0.0 AND 255.255.255.255) AND NOT
  (ipAddress between 0.0.0.0 AND 255.255.255.255))
  ```

or

```
((ipAddress not between ::ffff:0:0 AND ::ffff:ffff:ffff) AND NOT
(ipAddress between 0.0.0.0 AND 255.255.255.255))
```

- (*NNMi Advanced with IPv6 enabled*) All interfaces that have *any* IPv6 addresses (could also have IPv4)
  [click here for details of this filter.]

  The following example interface Group's Additional Filter finds any interface that has at least one IPv6 address:

  ```
  ((ipAddress between ::0 AND ::fffe:ffff:ffff) OR (ipAddress
  ::1:0:0:0 AND ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff))
  ```

- (*NNMi Advanced with IPv6 enabled*) All interfaces that have *both* IPv4 and IPv6 addresses (also known as dual-stack interfaces)
  [click here for details of this filter.]

  The following example interface Group's Additional Filter finds any interface that has at least one IPv4 address and at least one IPv6 address:

  ```
  ((ipAddress between 0.0.0.0 AND 255.255.255.255) AND (ipAddress not
  between 0.0.0.0 AND 255.255.255.255))
  ```

Note: To maximize the performance of Additional Filters based on an IP Address range, avoid multiple filter expressions. For example, use the `between` operator instead of the greater than or equal to (>=) and less than or equal to (<=) operators that cause NNMi to use multiple queries for finding all addresses that match the filter.

Guidelines for Creating Additional Filters for Interface Groups

The Additional Filters Editor enables you to create expressions to further define the interfaces to be included in an Interface Group. Make sure to design any complex Additional Filters offline as a Boolean expression first. This method can help to minimize errors when entering your expressions using the Additional Filters Editor.

When creating any Additional Filters for an Interface Group, note the following:

- Each set of expressions associated with a Boolean Operator is treated as if it were enclosed in parentheses and evaluated together. View the expression displayed under **Filter String** to see the logic of the expression as it is created.
- When using the AND operator to combine expressions that include Custom Attributes, include only one customAttrName/customAttrValue pair in a sub-expression.
- The AND and OR Boolean Operators must contain at least two expressions as shown in the example below.

  ```
  AND
     ifName like ATMS*
     ifName != ATMS/0/A
  ```

```
        OR
           ifSpeed = 10000000
           ifSpeed = 100000000
```

Note: As shown in the example above, you must use the actual ifSpeed number.

NNMi evaluates the expression above as follows:

```
(ifName like ATMS* AND ifName != ATMS/0/A) AND (ifSpeed = 10000000
OR ifSpeed = 100000000)
```

- NNMi finds all interfaces with an (interface name) ifName that begins with **ATMS**, but does not include **ATMS/0/A**.
- Of these interfaces, NNM then finds all interfaces with an (interface speed) ifSpeed of **10 Mbps** or **100 Mbps**.

- The placement of your cursor and the subsequent text that is selected is important when performing operations using the Additional Filters Editor. For example, you append to or replace, the expression that is selected.
- The placement of your cursor and the subsequent text that is selected is especially important when adding your Boolean operators. See "Add Boolean Operators in the Additional Filters Editor" on page 310 for more information.
- You can drag any of the following items to a new location in the Filter String:
  - Filter Editor Options: AND, OR, NOT, EXISTS, NOT EXISTS
  - Filter Expression (Attribute, Operator and Value)
- When moving items in the Filter String, note the following:
  - Click the item you want to move before dragging it to a new location.
  - As you drag a selected item, an underline indicates the target location.
  - If you are moving the selection up, NNMi places the item above the target location.
  - If you are moving the selection down, NNMi places the item below the target location.
  - If you attempt to move the selection to an invalid target location, NNMi displays an error message.

Add New ifType Values (Interface Types) to the List

Interface Type definitions cover all known industry-standard IANA ifType-MIB values at the time of the release of NNMi. Interface Groups can be built using `ifType` filters. See "Create Interface Groups" on page 321

Occasionally new industry-standard `ifType` values are announced between releases of NNMi. If your team acquires new devices configured with new `ifType` values, you can add the new `ifType` values to NNMi's list of definitions.

When NNMi discovers an Interface that responds to an SNMP `ifType` query with a new value, NNMi automatically adds a new ifType using the IANA ifType-MIB Number value. NNMi uses that number for both the `ifType` attribute and the Number attribute values. You can provide a more meaningful `ifType` text string and optional description.

To configure an IANA ifType-MIB definition:

1. Navigate to the **ifTypes** view:
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Expand **MIBs**.
   c. Select the **ifTypes** view.
2. Do one of the following:
   - To create an `ifType` definition, click the ✱ New icon, and continue.
   - To edit an `ifType` definition, click the Open icon in the row representing the configuration you want to edit, and continue.
   - To delete an `ifType` definition, select a row and click the ✖ Delete icon.
3. In the ifType form, provide the `ifType` text string, number, and description.
4. Click **Save and Close**.

In a CSV File, Define Interface Groups

Interface Groups are used for a variety of purposes in NNMi. See "Creating Groups of Nodes or Interfaces" on page 294 for more information.

You can create any number of Interface Groups in addition to the ones that NNMi provides (see "Interface Groups Provided by NNMi" on page 338).

You can create an Interface Group by either using the NNMi console or a comma separated values (CSV) file. For example, if you have Interface Group information in a Microsoft Excel spreadsheet, you can save this information as a .csv file and use the `nnmloadinterfacegroups.ovpl` command to add this interface group information to NNMi.

To create an Interface Group using a comma separated values (CSV) text file, use the `nnmloadinterfacegroups.ovpl` command:

Tip: See the nnmloadinterfacegroups.ovpl Reference Page for more information about the `nnmloadinterfacegroups.ovpl` command, including requirements for the CSV file. You must provide a CSV file with a specific syntax and order. Each column in the CSV file has a pre-defined meaning as described in the nnmloadinterfacegroups.ovpl Reference Page.

Here is an example syntax:

```
nnmloadinterfacegroups.ovpl -r [true|false] -u <NNMiadminUsername> -p
<NNMiadminPassword> -f <CSV file name>
```

CSV file name is the name of the CSV file that contains the Interface Group information.

`-r true` means *all the settings* for any existing Interface Group with the same `Name` are overwritten with the values in your CSV file.

Note: This is not a merge. It is a complete replacement of that Interface Group configuration.

`-r false` (defautl) means if the Interface Group `Name` already exists, the `nnmloadinterfacegroups.ovpl` command does not change the previous settings.

To create Node Groups using a CSV file, see "In a CSV File, Define Node Groups" on page 316

Troubleshooting Interface Changes

If your Interface Group definition results in unexpected membership or the membership changes, consider the strategy NNMi uses to detect Interfaces during Spiral Discovery.

During each Spiral Discovery cycle, NNMi responds to Interface changes as follows:

1. NNMi updates the attribute value of the current Interface object if one (*and only one*) of the following attributes change:
   - `ifIndex` or `IfAlias` or `ifSpeed`
2. NNMi creates a new Interface object and deletes the old Interface object if any of the following criteria are met:
   a. At least one of these attributes change: `ifName`, `ifDescriptions`, `ifType`, or Physical Address (Mac address, Media Access Control address).

   b. More than one of these attributes change: `ifIndex` or `IfAlias` or `ifSpeed`.

   c. One or more attributes from the list of both criteria 1 & 2 change.

> **Note:** If using nnmconnedit.ovpl configuration files, any connection settings configured for the deleted Interface would be evaluated for the new Interface object’s current attribute settings.

Node Groups Provided by NNMi

NNMi Provides the following kinds of Node Groups:

- Node Groups as Predefined View Filters. These Node Groups can also be used for Monitoring Configuration if you find them useful.
- "Island Node Groups" on page 337. NNMi automatically creates Island Node Groups whenever it detects changes in Layer 2 Connections. An Island Node Group is a group of fully-connected nodes that NNMi displays in a group that is not connected to the rest of the topology.

Node Groups As Predefined View Filters

NNMi provides the following Node Groups. You can configure these Node Groups with specific information about your management domain and change them to meet your needs.

Node Groups can be used to filter table views and map views, used for multiple configuration tasks, and exported to HP Network Node Manager iSPI Performance for Metrics Software and HP Network Node Manager iSPI Performance for Traffic Software.

Node Groups Provided by NNMi

Name	Purpose
Important Nodes	**Caution**: Do not delete this Node Group. This Node Group is used by the Causal Engine. Any devices in this group receive special treatment. When a current member of this group stops responding, the Causal Engine generates a "Node Down" incident and sets the device status to Critical. For example, when a WAN Edge Device is in the shadow of another problem (and, therefore, NNMi would normally not generate an incident about that WAN edge router), NNMi generates a "Node Down" incident because the router is listed in this Important Nodes group. This Node Group is empty by default. Consider populating this group with critical servers that run important applications and critical WAN routers. (*NNM iSPI Performance*) This group automatically becomes a filter for Performance Reports (unless the group has no members). The NNMi administrator can change this default behavior. See "Create Node Groups Using Filters or Hostname Lists (Configuration: Node Groups)" on page 296.
Microsoft Windows Systems	This Node Group includes any device manufactured by Microsoft. The Node Group definition is populated with one vendor entry. Any Microsoft devices within your management domain are automatically included in this Node Group.
Networking Infrastructure Devices	This Node Group is populated with a list of categories for network devices. Any devices within your management domain that match these categories are automatically included in this Node Group. Devices in this group are automatically monitored for Node Component fault metrics. (*NNM iSPI Performance*) This group automatically becomes a filter for Performance Reports (unless the group has no members). The NNMi administrator can change this default behavior. See "Create Node Groups Using Filters or Hostname Lists (Configuration: Node Groups)" on page 296. (*HP Network Node Manager iSPI Network Engineering Toolset Software*) By default, NNMi automatically uses NNM iSPI NET diagnostic flows to monitor devices in this group.
Non-SNMP Devices	This Node Group includes any device that does not respond to SNMP. The Node Group definition is populated with one entry for a null MIB-II `sysObjectID` value. Any device within your management domain that fails to respond to SNMP queries is automatically included in this Node Group.
Routers	This Node Group is populated with a list of categories for network devices that represent routers. Any router, switch-router, or gateway within your management domain is included in this Node Group. See Node Capabilities Provided by NNMi for more information. This filter is used to create the Routers Node Group map that NNMi provides by default in the Topology Maps workspace.

Node Groups Provided by NNMi, continued

Name	Purpose
	Devices in this group are automatically monitored for Node Component fault metrics (*NNM iSPI Performance*) Devices in this group are automatically monitored for performance, including Node Component performance metrics . This group automatically becomes a filter for Performance Reports. The NNMi administrator can change this default behavior. See "Default Settings for Monitoring" on page 345, "Node Group Settings for Monitoring" on page 391, and "Create Node Groups Using Filters or Hostname Lists (Configuration: Node Groups)" on page 296 for more information.
Switches	This Node Group is populated with a list of categories for network devices that represent switches. Any switch, ATM switch, or switch-router within your management domain is included in this Node Group. See Node Capabilities Provided by NNMi for more information. This filter is used to create the Switches Node Group map that NNMi provides by default in the Topology Maps workspace.

Node Groups Provided by *NNMi Advanced*

Name	Purpose
Virtual Machines	*NNMi Advanced*. Virtual machines being hosted on a VMware ESX/ESXi server. These servers are identified by a `com.hp.nnm.capability.node.VM` capability.
VMware ESX Hosts	*NNMi Advanced*. A VMware ESX/ESXi server that is hosting virtual machines. These servers are identified by a `com.hp.nnm.capability.node.hypervisor.vmware.ESX` capability.

Related Topics

"Island Node Groups" below (dynamically generated Node Groups)

Island Node Groups

An Island Group is a group of fully-connected nodes discovered by NNMi, and NNMi determines this group is not connected to the rest of the topology.

An example of an environment with multiple Island Node Groups is a financial institution or retail store with many branches or stores. Each branch or store might be connected to other branches or stores with a WAN (Wide Area Network) connection. Each branch or store appears as an isolated island of nodes in the NNMi topology.

NNMi automatically updates Island Node Group discovery information whenever it detects changes in Layer 2 Connections. NNMi begins rediscovery of the Island Node Group within a range of 10 seconds to 10 minutes, depending on current network traffic volume. NNMi uses the Discovery Interval to determine when the updates occur.

Note the following about Island Node Groups:

- NNMi selects a representative node in each Island Node Group as the Source Node associated with an Island Node Group incident. The representative node is selected using the following criteria:
  - Sort all routers in the Node Group alphabetically by name and choose the first one in the list
  - If no routers are in the Node Group, sort all nodes in the Node Group alphabetically by name and choose the first one in the list.
- Island Node Groups are identified using "Island" in the Node Group Name. NNMi also assigns each Island Node Group name a number to ensure the name is unique.
- Island Node Groups are manage internally. Therefore, NNMi administrators should not modify Island Node Group configurations. NNMi overrides any user changes the next time NNMi updates the Island Node Group discovery information.
- Island Node Groups must have at least two nodes.
- How the Status of Island Node Groups is calculated cannot be changed.

The only possible Status values for Island Node Groups are Unknown and Normal. Unknown indicates that NNMi cannot reach any nodes in the group. Normal indicates that NNMi can reach at least one node in the group.

Related Topics

"Node Groups As Predefined View Filters" on page 335

Interface Groups Provided by NNMi

NNMi Provides the following Interface Groups as predefined view filters. These Interface Groups can also be used for Monitoring Configuration if you find them useful.

Feel free to populate these Interface Groups with specific information about your management domain and change them to meet your needs.

Interface Groups Provided by NNMi

Name	Purpose
ATM Interfaces	This Interface Group includes all Interfaces identified as Asynchronous Transfer Mode (ATM) links. These Interfaces use a cell-based switching technique using asynchronous time division multiplexing.
DSx Interfaces	This Interface Group includes all Interfaces identified as Digital signal 1 (DS1, also known as T1) links. These Interfaces use a T-carrier signaling scheme to transmit voice and data between devices. Digital Signal 3 (DS3, also known as T3) links use a digital signal level 3 T-carrier.
Frame Relay Interfaces	This Interface Group includes all Interfaces identified as Frame Relay links. These Interfaces use a standardized wide area networking technology that specifies the physical and logical link layers of digital telecommunications channels using a packet switching methodology.
ISDN Interfaces	This Interface Group includes multiple Interface types known to be commonly used for ISDN purposes. Any Interface within your management domain that meets the defined criteria is automatically included in this Interface Group.

Interface Groups Provided by NNMi, continued

Name	Purpose
Link Aggregation Interfaces	*NNMi Advanced*. **Link Aggregation**[1] protocols: This Interface Group includes all *Aggregator Interfaces*. Network administrators can configure multiple *Aggregation Member Interfaces* on a switch to behave as one, the Aggregator Interface. This technique uses multiple interfaces in parallel to increase bandwidth, increase the speed at which data travels, and increase redundancy. See the Interface Form: Link Aggregation tab's Help topic for more information about Interfaces with Capability set to Aggregator Interface.
Point to Point Interfaces	This Interface Group includes multiple Interface types known to be commonly used for point-to-point purposes. Any Interface within your management domain that meets the defined criteria is automatically included in this Interface Group.
SONET Interfaces	This Interface Group includes all Interfaces identified as Synchronous Optical Networking (SONET) or Synchronous Digital Hierarchy (SDH) links. These Interfaces use a standardized multiplexing protocol that transfers multiple digital bit streams over optical fiber using lasers or light-emitting diodes (LEDs).
Software Loopback Interfaces	This Interface Group includes any Interface with an `ifType`Number value of 24, software loopback from the IANA ifType-MIB. Any Interface within your management domain that meets this **loopback address**[2] criteria is automatically included in this Interface Group.
VLAN Interfaces	This Interface Group includes Interfaces of `ifType`Number value of 135. The NNMi default Monitoring Configuration settings enable fault monitoring for these Interfaces, but disable performance monitoring (because collection of performance data for VLAN Interfaces tends to be problematic).
Voice Interfaces	This Interface Group includes multiple interface types known to be commonly used for voice purposes. Any Interface within your management domain that meets the defined criteria is automatically included in this Interface Group.
WLAN Interfaces	This Interface Group includes all Interfaces identified as Wireless Local Area Network (WLAN) links. These Interfaces connect two or more devices using some wireless distribution method, and might provide a connection through an access point to the wider Internet.

[1] Protocols used on Switches to configure multiple Interfaces (Aggregation Member Interfaces) to function as if they were one (an Aggregator Interface). When two Aggregator Interfaces establish a connection, that connection is an Aggregator Layer 2 Connection. The Aggregator Layer 2 Connection appears on Layer 2 Neighbor View maps as a thick line with an Interface icon at each end (representing the Aggregator Interface). When you double-click the thick line, it converts into multiple thin lines representing the participating Aggregation Member Layer 2 Connections with their Aggregation Member Interfaces at each end of the lines.

[2] The address associated with the loopback interface. The loopback interface is a virtual interface on a device that provides a route for internal communication. Many vendors provide a specially configured loopback for management purposes. Exact details of how loopbacks are configured varies by vendor and model. See each device's documentation for details. NNMi identifies these loopback addresses by using ifType Number 24, softwareloopback from the IANA ifType-MIB.

Chapter 10

Monitoring Network Health

NNMi administrators control which network devices NNMi monitors. By monitoring only the devices that are important within your network environment, the amount of traffic generated by NNM is kept to a minimum. NNMi administrators can configure NNMi to check devices with status *other than critical* less frequently (if at all) to prevent unimportant incidents from showing up in the Incident views.

Before configuring NNMi monitoring behavior, the following tasks must be completed:

- "Configuring Communication Protocol" on page 119
- "Discovering Your Network" on page 175

For the most flexibility, also complete these tasks:

- Review the "Interface Groups Provided by NNMi" on page 338 and "Node Groups Provided by NNMi" on page 335.
- Create your own groups by "Creating Groups of Nodes or Interfaces" on page 294.

NNMi administrators configure NNMi monitoring behavior to meet your team's needs:

1. Start by establishing the appropriate settings for the monitoring tools provided by NNMi. See "Configure NNMi Monitoring Behavior" below.

   The State Poller and the Causal Engine work together to monitor the health of your network. Many of the tasks your team normally does to troubleshoot network problems can be automated. To learn more about how this works, see the following topics:

   - "About the State Poller" on page 342
   - "The NNMi Causal Engine and Monitoring" on page 343

2. Then write your own custom monitoring tools to meet any special requirements for your team. See "Create Custom Polling Configurations" on page 419.

Configure NNMi Monitoring Behavior

Certain devices in your network are the most important ones. You and your team must keep those devices up and running at all times. Adjust NNMi monitoring behavior to focus on the important devices and to check devices with status *other than critical* less frequently (if at all).

> **Note:** NNMi does not poll any private interface, IPv4 **Anycast Rendezvous Point IP Address**[1] or IPv6 Anycast address.

[1] Rendezvous Point addresses are loopback addresses used for routers in multi-cast network configurations.

Based on your individual situation, adjust the NNMi behavior to meet your needs. NNMi applies your Monitoring Configuration settings in the following sequence:

1. **Interface Group Settings**: NNMi monitors each of the Node's Interfaces and IP Addresses based on the first matching Interface Group Settings definition. The first match is the Interface Group Settings definition with the lowest Ordering number, then Baseline Settings.
2. **Node Group Settings**: NNMi monitors each Node and each previously unmatched Interface or IP Address based on the first matching Node Group Settings definition. The first match is the Node Group Settings definition with the lowest Ordering number, then Baseline Settings.

   > **Note:** Child node groups are included in the Ordering hierarchy. This means that if the parent node group has a lower Ordering number (for example, parent=10, child=20), then the monitoring configuration specified for the parent node group also applies to the nodes in the child node group. To override a parent node group monitoring configuration, set the Ordering number for the child node group to a number that is lower than the parent (for example, parent=20, child=10).

3. **Default Settings**: If no match is found for a Node, Interface, or IP Address in 1 or 2, NNMi applies the default Monitoring Configuration settings.

Tasks for Configuring the Monitoring Behavior

Task	How
"Global Control Settings for Monitoring" on page 343.	*Optional*. Use the Global Control group.
"Default Settings for Monitoring" on page 345.	Use the Default Settings tab to establish monitoring behavior for any devices that are discovered, but not included in any Node Group Settings or Interface Group Settings definitions.
"Interface Group Settings for Monitoring (NNM iSPI Performance for Metrics)" on page 374.	*Optional*. Use the Interface Group Settings tab. Configure settings based on Interface Groups to customize the way NNMi monitors certain groups of interfaces in your environment. Prerequisite: "Create Interface Groups" on page 321.
"Node Group Settings for Monitoring" on page 391.	*Optional*. Use the Node Group Settings tab. Configure settings based on Node Groups to customize the way NNMi monitors certain groups of devices in your environment. Prerequisite: "Create Node Groups" on page 295.
"Detect Interface Changes" on page 280.	*Optional*. Use Device Profiles to configure how NNMi detects interface changes.
"Monitor Router Redundancy Groups (NNMi Advanced)" on page 415.	*Optional*. Use additional settings to fine tune the way NNMi monitors Router Redundancy Groups.

About the State Poller

The State Poller Service monitors each discovered interface, address, card, and SNMP agent that is designated to be actively monitored in your management domain. State Poller can also be configured to provide Node Component monitoring and Router Redundancy Group monitoring.

State Poller gathers information in the following area and updates the ***State*** field on each object's form:

- Verifies that each monitored IP Address is responding to ICMP ping.
- Verifies that each monitored SNMP Agent is responding to SNMP queries.
- Issues SNMP queries for the following:
    - Each monitored interface, requesting the current value for MIB-II `ifAdminStatus` and `ifOperStatus`. (`ifAdminStatus` is set by the device administrator. `ifOperStatus` indicates the operational status of interface health.)
    - Router Redundancy Groups.
    - Node Component data.
- By default, State Poller monitors interfaces connected to another known interface through a Layer 2 Connection.
- You can extend monitoring to include the following:
    - Unconnected interfaces
    - Interfaces that have an IP address (for example a router interface that services mobile laptop machines)
    - (*NNM iSPI Performance for Metrics*). The State Poller also collects performance data and monitors thresholds. See "Purchase an HP Network Node Manager i Smart Plug-in" on page 1486.

The State Poller stores the State changes resulting from the queries in the NNMi database and notifies the Causal Engine of any changes. When notifying the Causal Engine of any changes, the State Poller sends only those State values that have changed.

> **Tip:** To force the State Poller to send the Causal Engine all of the State information it can collect regardless of changes, use **Actions → Status Poll** or the `nnmstatuspoll.ovpl` command. See Verify the Current Status for a Device for more information about Status Poll.

The Causal Engine gathers additional information about the overall health of each interface and SNMP agent. Using the State information collected from the State Poller as well as this additional information the Causal Engine calculates the ***Status*** of each node, interface, and SNMP agent.

> **Note:** Any time the State Poller sends updated State values for a selected object, the Causal Engine reanalyzes Status, Conclusions, and Incidents, and updates this information if needed.

See "The NNMi Causal Engine and Monitoring" on the next page for more information.

To configure the behavior of the State Poller, see:

- "Default Settings for Monitoring" on page 345
- "Global Control Settings for Monitoring" below
- "Configure Default SNMP, Management Address, and ICMP Settings" on page 120

The NNMi Causal Engine and Monitoring

The Causal Engine actively gathers information about your network devices from incoming incidents and traps. The Causal Engine also uses the data gathered by State Poller and by Spiral Discovery to calculate the current health status of each managed object.

The health status is dynamic (based on the current reality of your network environment). Any time the State Poller sends updated State values for an object, the Causal Engine reanalyzes Status, Conclusions, and Incidents, and updates this information if needed.

Note: The Causal Engine performs a Status Poll of each node every 24 hours and updates Status, Conclusion, and Incident information as needed. This Status Poll does not affect the timing of the Polling interval configured for the device.

The NNMi Causal Engine communicates device health information in the following ways:

- In the database, the Causal Engine stores a multitude of information about each device. You can access this information in the Node, Interface, IP Address, SNMP Agent, and connection forms.
- On the maps, the color of the background shape for each map icon changes to the color that represents the currently calculated health status, based on Causal Engine calculations for that node, interface, address, or connection (click here for information about status colors).
- On forms for Nodes, Interfaces, IP addresses, SNMP Agents, and connections, the Causal Engine updates the Status attribute to show the current status: ✓ **Normal**, ▲ **Warning**, ⚠ **Minor**, ▼ **Major**, ✖ **Critical**, ? **Unknown**, or ⊘ **No Status**.
- The Status column in table views is updated.

The Causal Engine also uses health status information to determine root cause. See "The NNMi Causal Engine and Incidents" on page 592 for more information about the Causal Engine, incidents, and root cause analysis.

Global Control Settings for Monitoring

Note: To suspend all SNMP traffic generated by NNMi, rather than only the State Poller Service SNMP traffic, see "Communication Region Form" on page 137 and "Specific Node Settings Form (Communication Settings)" on page 155.

To temporarily turn off all NNMi monitoring activity without tampering with your customized monitoring configuration settings:

1. Navigate to the **Monitoring Configuration** form.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Expand the **Monitoring** folder.

c. Select the **Monitoring Configuration**.

2. Locate the **Global Control** group box and for each setting do the following (see table):

   □ = disable

   ☑ = enable

3. Click **Save and Close**. NNMi applies your changes. The next regularly scheduled monitoring cycle uses the new settings.

Global Control

Name	Description
Enable State Polling	If ☑ enabled, State Poller monitors all managed objects (for example, interfaces, IP addresses, and SNMP agents) by issuing ICMP pings and SNMP read-only queries for MIB-II ifAdminStatus and ifOperStatus. (ifAdminStatus is set by the device administrator. ifOperStatus indicates the overall health of the device and is supplied by the SNMP Agent.) You can also configure NNMi so that State Poller gathers additional information about Node Components and Router Redundancy Groups. If □ disabled: • Previously discovered devices remain with the last calculated state/status. • Newly discovered devices are set to "No Status" with map-symbol background shape color set to beige.
Enable Card Polling	If ☑ enabled, State Poller monitors all managed cards. See Card Form for more information about card metrics. **Note:** Card monitoring is enabled by default. If □ disabled: • Previously discovered cards are assigned a State of **Not Polled** and a Status of **No Status** for Card metrics. • Newly discovered cards are assigned a State of **Not Polled** and a Status of **No Status**.
Enable Chassis Polling	If ☑ enabled, State Poller monitors all managed chassis. See Chassis Form for more information about chassis metrics. **Note:** Chassis monitoring is enabled by default. If □ disabled: • Previously discovered chassis are assigned a State of **Not Polled** and a Status of **No Status** for chassis metrics. • Newly discovered chassis are assigned a State of **Not Polled** and a Status of

Global Control , continued

Name	Description
	No Status.
Enable Node Component Polling	If ☑ enabled, State Poller monitors Node Component metrics for all managed nodes. See Node Form: Node Component Tab for more information about Node Component metrics. **Note:** Node Component monitoring is enabled by default. Only the health of Fan and Power Supply Node Components are propagated to the Node level. If □ disabled: • Previously discovered devices are assigned a State of **Not Polled** and a Status of **No Status** for Node Component metrics. • Node Component metrics for newly discovered devices are assigned a State of **Not Polled** and a Status of **No Status**.
Enable Router Redundancy Group Polling (*NNMi Advanced*)	If ☑ enabled, NNMi monitors all managed Router Redundancy Groups. See Router Redundancy Group View (*NNMi Advanced*) for more information about Router Redundancy Groups. **Note:** Router Redundancy Group monitoring is enabled by default. If □ disabled: • Previously discovered Router Redundancy Groups are assigned a State of **Not Polled** and a Status of **No Status**. • Newly discovered Router Redundancy Groups are assigned a State of **Not Polled** and a Status of **No Status**.

Default Settings for Monitoring

The choices you make for "defaults" apply only to devices with interfaces, IP addresses, cards, SNMP agents (Management Addresses), Tracked Objects, Router Redundancy Groups, or Node Components that are not covered by any monitoring Interface Group Settings or Node Group Settings.

To establish default NNMi monitoring behavior:

1. Navigate to the **Defaults Settings** tab.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Expand the **Monitoring** folder.
   c. Select **Monitoring Configuration**.
   d. Locate the **Defaults Settings** tab.

2. To prevent NNMi from generating any SNMP traffic to Nodes that are not covered by Monitoring Configuration's Node Group Settings. See Enable SNMP Polling on Node.

3. Locate the **Default Fault Monitoring** group box.

   a. Configure the Default Fault Monitoring behavior for ICMP traffic (see Default Fault Monitoring: ICMP Fault Monitoring table).

   b. Configure the Default Fault Monitoring behavior for SNMP traffic (see Default Fault Monitoring: SNMP Fault Monitoring table).

   c. Configure the Default Fault Monitoring: interval (see Default Fault Monitoring: interval table).

4. (*NNM iSPI Performance for Metrics*) If the HP Network Node Manager iSPI Performance for Metrics Software is installed, locate the **Default Performance Monitoring** group box.

   Configure the Default Performance Monitoring behavior (see and Default SNMP Performance Monitoring table).

5. By default, NNMi monitors only interfaces that are connected to other interfaces. When SNMP polling is enabled, NNMi automatically detects most connections. See "Add or Delete a Layer 2 Connection" on page 284 for information about manual overrides.

   Optional. If you want to expand default monitoring behavior to include unconnected Interfaces, indicate your choices in the Default Extend the Scope of Polling Beyond Connected Interfaces group box

6. *Optional*. Configure the Default Change Detection Monitoring (see Default Change Detection Monitoring table).

7. *Optional*. To establish custom monitoring behavior for one or more groups of interfaces, configure Interface Group Settings, see "Interface Group Settings for Monitoring (NNM iSPI Performance for Metrics)" on page 374.

8. *Optional*. To establish custom monitoring behavior for one or more groups of nodes, configure Node Group Settings, see "Node Group Settings for Monitoring" on page 391.

9. Click **Save and Close**. NNMi applies your changes. The next regularly scheduled monitoring cycle uses the new settings.

   > **Caution:** When you establish monitoring configuration settings, NNMi must recalculate the settings for all affected objects. This can take some time and slow down your system. Consider making this change during a slow time in your network environment.

Default Monitoring

Attribute	Description
Enable SNMP Polling on Node	If ☑ enabled, NNMi contacts the SNMP Agent on each node in your network to gather SNMP data for monitoring purposes (unless the Monitoring Configuration's Node Group Settings specifically disables SNMP for monitoring purposes). If □ disabled, NNMi does not contact the SNMP Agent on this node for monitoring purposes (does not generate SNMP traffic to the node).

Default Monitoring, continued

Attribute	Description
	Note: If you use Auto-Discovery, NNMi might detect Nodes and add them to the NNMi database as non-SNMP nodes. To configure Auto-Discovery to not add specified IP addresses to the NNMi database, not acknowledge any Hints received about them, nor gather Discovery Hints from them unless the address is a discovery seed, see "Set Outside Limits for Auto-Discovery" on page 228.

Default Fault Monitoring: ICMP Fault Monitoring

Attribute	Description
Enable Management Address Polling	If ☑ enabled, State Poller only issues ICMP (ping) requests to the management address for a node. **Note:** In the Global Control section of the Monitoring Configuration form, the Enable State Polling attribute must be enabled, too. If □ disabled, State Poller does one of the following: • If neither this attribute nor *Enable ICMP Fault Polling* is selected, State Poller does not use ICMP to monitor nodes covered by this configuration setting. • If *Enable ICMP Fault Polling* is selected, State Poller uses ICMP to monitor ALL IP addresses covered by this configuration setting. Changing the default monitoring settings for the management addresses takes effect immediately. To verify the change, see "Verify the Monitoring Settings" on page 416.
Enable IP Address Fault Polling **Note:** This monitoring option is useful for devices that do not support SNMP. By default, this feature is enabled for the "Non-SNMP Devices" Node Group.	If ☑ enabled, State Poller issues ICMP (ping) requests to verify the availability of discovered IP address. **Note:** In the Global Control section of this form, the Enable State Polling attribute must be enabled, too. If □ disabled, State Poller does the following: • If neither this attribute nor *Management IP Address Polling* is selected, State Poller does not use ICMP to monitor nodes covered by this configuration setting. • IP addresses (both previously discovered and newly discovered) have a State attribute value of "Not Polled" and a Status attribute value of "No Status" with the color of the IP address map-symbol set to beige. See Layer 3 Neighbor View.

Default Fault Monitoring: ICMP Fault Monitoring, continued

Attribute	Description
	• If both ICMP and SNMP are disabled for a Node, the Node has a Status attribute value of "No Status" and the color of the Node map-symbol background shape is set to beige. **Tip:** To turn off ICMP polling within a subset of your network environment, use the Communication Configuration workspace Region definitions. You can define your own Regions that identify any unreachable addresses in your management domain (for example, the **private IP addresses**[1]).

Default Fault Monitoring: SNMP Fault Monitoring

Attribute	Description
Enable Interface Fault Polling	If ☑ enabled, State Poller monitors all interfaces by issuing SNMP read-only queries to devices assigned to this level of the monitoring hierarchy. By default, any connected interface is monitored for MIB-II ifAdminStatus and ifOperStatus. (ifAdminStatus is set by the device administrator. ifOperStatus indicates the operational status of interface health.) If you have *unconnected* interfaces that you want to monitor, expand NNMi monitoring behavior with the Poll Unconnected Interfaces and the Poll Interfaces Hosting IP Addresses attributes. **Note:** The following attributes must also be enabled: • In the Global Control section of this form, the Enable State Polling attribute must be enabled, too. See Layer 2 Neighbor View. (See "Global Control Settings for Monitoring" on page 343 for more information.) • In the Communication Configuration view, enable State Poller queries with the applicable Enable SNMP Communication attributes (see "Configuring Communication Protocol" on page 119 for more information). If □ disabled, for devices assigned to this level of the monitoring hierarchy: • Causal Engine calculates Status based only on IP address State. • The Interface objects previously discovered change to a State attribute value of "Not Polled" and a Status attribute value of "No Status" (plus

[1] These are IPv4 addresses that can be reused in home and office local area networks (LANs). Following the standards set by RFC 1918 and RFC 4193 (10.*.*.*, 169.254.*.*, 172.16-31.*.*, and 192.168.*.*)

Default Fault Monitoring: SNMP Fault Monitoring, continued

Attribute	Description
	any related map-symbol changes to a beige color).
Enable Card Fault Polling	Use this attribute to poll fault metrics for cards. Card fault metrics include Administrative State, Operational State, and Standby State. **Note:** Card Fault Polling is enabled by default. If ☑ enabled, NNMi gathers fault data related to the card fault metrics in devices assigned to this level of the monitoring hierarchy. If ☐ disabled, NNMi does not extend data collection behavior to include card fault data about devices assigned to this level of the monitoring hierarchy. **Tip:** NNMi uses the same polling interval set for the Fault Polling Interval.
Enable Node Component Fault Polling **Note:** By default, this feature is enabled for the "Routers" and "Networking Infrastructure Devices" Node Groups.	Use this attribute to poll Node Component fault metrics. Node Component fault metrics include the following: Fan, Power Supply, Temperature, and Voltage. **Note:** Node Component Fault Polling is disabled by default. Only the health of the Power Supply and Fan Node Components are propagated to the Node level. If ☑ enabled, NNMi gathers fault data related to the Node Component fault metrics in devices assigned to this level of the monitoring hierarchy. If ☐ disabled, NNMi does not extend data collection behavior to include Node Component fault data about devices assigned to this level of the monitoring hierarchy. **Tip:** NNMi uses the same polling interval set for the Fault Polling Interval.

Default Fault Monitoring: interval

Attribute	Description
Fault Polling Interval	The time that State Poller waits between issuing queries to gather information for any of the following that are enabled: ICMP Polling, SNMP Polling, Poll Unconnected Interfaces, and Poll Interfaces Hosting IP addresses. The default Fault Polling Interval is 5 minutes, except for the Node Group named Microsoft Windows Systems which is 10 minutes.

Default Fault Monitoring: interval, continued

Attribute	Description
	Note: NNMi monitors SNMP agents (Management Addresses) according to this Fault Polling Interval, *even if ICMP Polling, SNMP Polling, Poll Unconnected Interfaces, and Poll Interfaces Hosting IP addresses are all disabled*. To prevent an SNMP Agent's address from being monitored, one of the following must be true: State Polling is disabled, current Communication Configuration settings turn off SNMP for the SNMP agent's address, the parent Node is set to Not Managed or Out of Service, or the parent node belongs to a Monitoring Configuration's Node Group with □ **Enable SNMP Polling on Node** disabled.

Default SNMP Performance Monitoring (*NNM iSPI Performance for Metrics*)

Attribute	Description
LAN Performance Monitoring: Enable Interface Performance Polling	(*NNM iSPI Performance for Metrics*) Use this attribute to extend the range of polling data that NNMi collects. HP Network Node Manager iSPI Performance for Metrics Software uses the additional data in a series of performance reports. See "Purchase an HP Network Node Manager i Smart Plug-in" on page 1486 for more information. When enabled, network traffic increases on your network because NNMi gathers performance data about each member of this group on a regular schedule. If ☑ enabled, NNMi gathers basic Interface performance data from Interfaces in devices assigned to this level of the monitoring hierarchy. If □ disabled, NNMi does not extend data collection behavior to include performance data about Interfaces assigned to this level of the monitoring hierarchy. **Note:** The Enable State Polling field must be enabled, too. By default the performance of connected interfaces and addresses is monitored. If you have *unconnected* interfaces that you want to monitor, expand NNMi monitoring behavior by enabling Poll Unconnected Interfaces.
WAN Performance Monitoring: Enable DSx Interface Performance Polling	(*NNM iSPI Performance for Metrics*) Use this attribute to extend the range of polling data that NNMi collects. HP Network Node Manager iSPI Performance for Metrics Software uses the additional data in a series of performance reports. When enabled, network traffic increases on your network because NNMi gathers performance data about each member of the **DSx Interfaces** interface group on a regular schedule. See "Interface Groups Provided by NNMi" on page 338 for more information. If ☑ enabled, NNMi gathers DSx performance data from DSx Interfaces assigned to this level of the monitoring hierarchy. If □ disabled, NNMi does not gather DSx performance data from DSx Interfaces assigned to this level of the monitoring hierarchy.

Default SNMP Performance Monitoring (NNM iSPI Performance for Metrics), continued

Attribute	Description
WAN Performance Monitoring: Enable SONET Interface Performance Polling	(*NNM iSPI Performance for Metrics*) Use this attribute to extend the range of polling data that NNMi collects. HP Network Node Manager iSPI Performance for Metrics Software uses the additional data in a series of performance reports. When enabled, network traffic increases on your network because NNMi gathers performance data about each member of the **SONET Interfaces** interface group on a regular schedule. See "Interface Groups Provided by NNMi" on page 338 for more information. If ☑ enabled, NNMi gathers SONET performance data from SONET Interfaces assigned to this level of the monitoring hierarchy. If □ disabled, NNMi does not gather SONET performance data from SONET Interfaces assigned to this level of the monitoring hierarchy.
WAN Performance Monitoring: Enable ATM Interface Performance Polling	(*NNM iSPI Performance for Metrics*) Use this attribute to extend the range of polling data that NNMi collects. HP Network Node Manager iSPI Performance for Metrics Software uses the additional data in a series of performance reports. When enabled, network traffic increases on your network because NNMi gathers performance data for each ATM Interface. If ☑ enabled, NNMi gathers ATM performance data from ATM Interfaces assigned to this level of the monitoring hierarchy. If □ disabled, NNMi does not gather ATM performance data from ATM Interfaces assigned to this level of the monitoring hierarchy. **Note:** • This option gathers metrics from ATM-MIB and CISCO-AAL5-MIB. • See also "Configure Discovery of ATM/Frame Relay Interfaces" on page 201.
WAN Performance Monitoring: Enable Frame Relay Interface Performance Polling	(*NNM iSPI Performance for Metrics*) Use this attribute to extend the range of polling data that NNMi collects. HP Network Node Manager iSPI Performance for Metrics Software uses the additional data in a series of performance reports. When enabled, network traffic increases on your network because NNMi gathers performance data for each Frame Relay Interface. If ☑ enabled, NNMi gathers Frame Relay performance data from Frame Relay Interfaces assigned to this level of the monitoring hierarchy. If □ disabled, NNMi does not gather Frame Relay performance data from Frame Relay Interfaces assigned to this level of the monitoring hierarchy. This option gathers the following types of metrics: • Circuit in and out octets, errors, and discards

Default SNMP Performance Monitoring (NNM iSPI Performance for Metrics), continued

Attribute	Description
	• Committed Information Rate (CIR) and Extended Information Rate (EIR) utilization • Forward Error Congestion Notification (FECN) and Backward Error Congestion Notification (BECN) counts See also "Configure Discovery of ATM/Frame Relay Interfaces" on page 201.
Enable Node Component Performance Polling **Note:** By default, this feature is enabled for the **Routers** Node Group if NNM iSPI Performance for Metrics is installed.	(*NNM iSPI Performance for Metrics*) Use this attribute to poll Node Component performance. An NNMi administrator can set the threshold for node components related to the following performance metrics: CPU utilization, memory utilization, buffer utilization, buffer miss rate, and buffer failure rate. **Note:** Node Component Performance Polling is disabled by default. If ☑ enabled, NNMi gathers performance data related to the Node Component performance metrics in devices assigned to this level of the monitoring hierarchy. If □ disabled, NNMi does not extend data collection behavior to include Node Component performance data about devices assigned to this level of the monitoring hierarchy. **Tip:** NNMi uses the same polling interval set for the Performance Polling Interval.
Performance Polling Interval	(*NNM iSPI Performance for Metrics*) Use this field to set the time period that NNMi waits between issuing network traffic to gather performance data for the HP Network Node Manager iSPI Performance for Metrics Software. The default Performance Polling Interval is 5 minutes, except for the Node Group named Microsoft Windows Systems which is 10 minutes.

Default Extend the Scope of Polling Beyond Connected Interfaces

Attribute	Description
Poll Unconnected Interfaces	If ☑ enabled, NNMi monitors all interfaces within discovered devices (both connected and unconnected). All interfaces are monitored for MIB-II ifAdminStatus and ifOperStatus. (ifAdminStatus is set by the device administrator. ifOperStatus indicates the operational status of interface health.) **Note:** The Enable State Polling field must be enabled, and

Default Extend the Scope of Polling Beyond Connected Interfaces , continued

Attribute	Description
	SNMP polling of some type must be enabled (for example, Enable SNMP Fault Monitoring and Enable SNMP Performance Polling). If □ disabled, State Poller polls according to other configuration settings. **Tip:** Your discovery configuration choices might need to be adjusted to get the results you want. For example, to meet the “connected” criteria for interfaces in switches that do not have an IP address you must add the device to which the interface is connected as a discovery seed. See"Specify Discovery Seeds" on page 256.
Poll Interfaces Hosting IP Addresses **Note:** This monitoring option is useful for Router interfaces. By default, this feature is enabled for the "Routers" Node Group.	If ☑ enabled, any unconnected interface that has one or more addresses associated with it is monitored for MIB-II ifAdminStatus and ifOperStatus. (ifAdminStatus is set by the device administrator. ifOperStatus indicates the operational status of interface health.) **Note:** The Enable State Polling field must be enabled, and SNMP polling of some type must be enabled (for example, Enable SNMP Fault Monitoring and Enable SNMP Performance Polling). By monitoring the Interface (in addition to the IP address), NNMi can make more informed decisions about the health of each IP address associated with an unconnected interface. If □ disabled, State Poller polls according to other configuration settings. **Tip:** The Communication Configuration workspace provides a method of overriding this setting for specific Regions. You can define your own Region to easily turn off polling to any unreachable addresses in your management domain (for example, the **private IP addresses**[1]).

[1] These are IPv4 addresses that can be reused in home and office local area networks (LANs). Following the standards set by RFC 1918 and RFC 4193 (10.*.*.*, 169.254.*.*, 172.16-31.*.*, and 192.168.*.*)

Default Change Detection Monitoring

Attribute	Description
Enable Number of Interfaces (ifNumber) Polling	**Tip:** For more information, see "Detect Interface Changes" on page 280. When enabled ☑, NNMi polls for the number of interfaces using the `ifNumber` value for the node. If the number of interfaces has changed, NNMi initiates a rediscovery of the node. Polling is suspended until the discovery is complete. When disabled □, NNMi does not actively poll for a change in the number of interfaces. The change is detected the next time the node is rediscovered.
Enable Entity Change Time (entLastChangeTime) Polling	When enabled ☑, NNMi polls for the last change time from the ENTITY-MIB `entLastChangeTime` value. If the time has changed, NNMi initiates a rediscovery of the node. Polling is suspended until the discovery is complete. When disabled □, NNMi does not actively poll the `entLastChangeTime` MIB value. The change is detected the next time the node is rediscovered.
Change Detection Polling Interval	The time that State Poller waits between issuing queries to gather information for the Number of Interfaces (`ifNumber`) and Entity Change Time (`entLastChangeTime`) settings enabled for Change Detection Monitoring. The default Change Detection Polling Interval is 4 hours.

About Threshold Settings Provided by NNMi

Instruct NNMi to monitor thresholds for devices throughout your network (for example, Interface Input Utilization).You can also do the following when any of these thresholds are enabled:

- Configure incidents related to these thresholds, for more information:
- Configure custom incident attributes for these thresholds,for more information:
- There are many benefits to using these thresholds provided by NNMi:
  a. NNMi provides all the complex logic required to conduct the threshold evaluations, accessing the appropriate MIB to provide the most accurate data for each specific device. For more information:
     - In the NNMi console, click Help → NNMi Documentation Library → Release Notes.
     - Click the link to HP Network Node Manager i Software System and Device Support Matrix.
     - Click the link at the top of the file "For the latest additions to the system requirements and device support".

- Click the link to the **device support matrix**. Review the list of MIB files that NNMi is configured to use when appropriate.

b. NNMi gathers Monitored Attribute data, evaluates any established thresholds to determine State values for the devices you are monitoring.

> **Tip:** If these thresholds do not meet all of your team's needs, write your own. See "Configure Threshold Information for a Custom Poller Collection" on page 441 (your Custom Poller thresholds affect the Custom Polled Instance State, and can be configured to affect Node Status and generate associated incidents).

Available Threshold Attributes for Monitoring Configuration

	Relevant for:				
NNNMi Attributes available for Thresholds	**Interface Group: Threshold Settings**	**Interface Group: Baseline Settings**	**Node Group: Threshold Settings**	**Node Group: Baseline Settings**	
Order applied (low to high #):	**1st Group Order #s 1 - x**	**2nd Group Order #s 1 - x**	**3rd Group Order #s 1 - x**	**4th Group Order #s 1 - x**	**Controlled by:**
Management Address ICMP Response Time Time elapsed (in milliseconds) before receiving a reply to NNMi's Internet Control Message Protocol (ICMP) echo request. NNMi issues the ICMP echo request to the Node's management address.			X	X	Fault Polling

Prerequisite for ICMP monitoring:

1. Navigate to the **Node Group Settings** form.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Expand the **Monitoring** folder.
   c. Select **Monitoring Configuration**.
   d. Locate the **Node Group Settings** tab.
   e. Open the appropriate **Node Group Settings** form.
2. Scroll down to the **Fault Monitoring** section, locate the ICMP Fault Monitoring settings. The ☑ Enable Management Address Polling must be enabled.

Available Threshold Attributes for Monitoring Configuration, continued

	Relevant for:				
NNNMi Attributes available for Thresholds	**Interface Group: Threshold Settings**	**Interface Group: Baseline Settings**	**Node Group: Threshold Settings**	**Node Group: Baseline Settings**	
Order applied (low to high #):	**1st Group Order #s 1 - x**	**2nd Group Order #s 1 - x**	**3rd Group Order #s 1 - x**	**4th Group Order #s 1 - x**	**Controlled by:**
Tip: NNMi administrators can check network latency for a Node Group or Interface Group by adjusting the following for the management addresses associated with the specified group of nodes or interfaces: • ICMP polling frequency • ICMP echo request packet data payload size See "Maintaining NNMi" chapter in the *HP Network Node Manager i Software Deployment Reference* for more information.					

Threshold Settings for NNM iSPI Performance for Metrics

The rest of the NNMi-provided threshold settings require that HP Network Node Manager iSPI Performance for Metrics Software be installed. See "Purchase an HP Network Node Manager i Smart Plug-in" on page 1486 for more information. NNM iSPI Performance for Metrics provides exception reports to track the frequency of threshold breaches. You can open these reports with **Actions** → **HP NNM iSPI Performance** → **Reporting - Report Menu** in the incident, node, or interface views and forms. (See NNM iSPI Performance for Metrics Actions.)

The following thresholds apply to Nodes (see "Node Group Settings for Monitoring" on page 391):

Prerequisite for these Node Group Thresholds:

1. Navigate to the **Node Group Settings** form.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Expand the **Monitoring** folder.
   c. Select the **Monitoring Configuration** form.
   d. Locate the **Node Group Settings** tab.
   e. Open the appropriate **Node Group Settings** form.
2. Scroll down to the **SNMP Performance Monitoring** section, locate the Component Performance Monitoring settings. The ☑ Enable Node Component Performance Polling must be enabled.

Available Node Group Threshold Attributes for Monitoring Configuration

	Relevant for:				
NNM iSPI Performance for Metrics Monitored Attributes available for Thresholds	**Interface Group: Threshold Settings**	**Interface Group: Baseline Settings**	**Node Group: Threshold Settings**	**Node Group: Baseline Settings**	
Order applied (low to high #):	**1st Group Order #s 1 - x**	**2nd Group Order #s 1 - x**	**3rd Group Order #s 1 - x**	**4th Group Order #s 1 - x**	**Controlled by:**
Backplane Utilization Threshold based on the percentage of backplane usage compared to the total amount of available backplane resources.			X	X	Performance Polling
Buffer Failure Rate Threshold based on the percentage of a node's buffer failures compared to the total number of attempts to create new buffers. These failures are caused by insufficient memory when the device tried to create new buffers.			X		Performance Polling
Buffer Miss Rate Threshold based on the percentage of a Node's buffer misses compared to the total attempts at buffer access. Crossing this threshold indicates the number of available buffers are dropping below a minimum level required for successful operation.			X		Performance Polling
Buffer Utilization Threshold based on the percentage of a Node's buffers that are currently in use, compared to the total number of available buffers.			X	X	Performance Polling
CPU 1Min Utilization			X	X	Performance

Available Node Group Threshold Attributes for Monitoring Configuration, continued

	Relevant for:				
NNM iSPI Performance for Metrics Monitored Attributes available for Thresholds	**Interface Group: Threshold Settings**	**Interface Group: Baseline Settings**	**Node Group: Threshold Settings**	**Node Group: Baseline Settings**	
Order applied (low to high #):	**1st Group Order #s 1 - x**	**2nd Group Order #s 1 - x**	**3rd Group Order #s 1 - x**	**4th Group Order #s 1 - x**	**Controlled by:**
Threshold based on the percentage of a node's CPU usage compared to the total amount of available CPU capacity. This percentage is the average CPU utilization over the prior 1-minute.					Polling
CPU 5Min Utilization Threshold based on the percentage of a node's CPU usage compared to the total amount of available CPU capacity. This percentage is the average CPU utilization over the prior 5-minutes.			X	X	Performance Polling
CPU 5Sec Utilization Threshold based on the percentage of a node's CPU usage compared to the total amount of available CPU capacity. This percentage is the average CPU utilization over the prior 5-seconds.			X	X	Performance Polling
Disk Space Utilization Threshold based on the percentage of a node's disk space usage compared to the total amount of available disk space.			X	X	Performance Polling
Memory Utilization Threshold based on the percentage of a node's memory usage compared to			X	X	Performance Polling

Available Node Group Threshold Attributes for Monitoring Configuration, continued

	Relevant for:				
NNM iSPI Performance for Metrics Monitored Attributes available for Thresholds	**Interface Group: Threshold Settings**	**Interface Group: Baseline Settings**	**Node Group: Threshold Settings**	**Node Group: Baseline Settings**	
Order applied (low to high #):	**1st Group Order #s 1 - x**	**2nd Group Order #s 1 - x**	**3rd Group Order #s 1 - x**	**4th Group Order #s 1 - x**	**Controlled by:**
the total amount of available memory.					

The Monitored attributes in the following table are available as Interface Group thresholds and Node Group thresholds. For each monitored Interface, NNMi applies any relevant Interface Group threshold configurations first. If none are available, NNMi applies any relevant Node Group threshold configurations.

The following thresholds apply to Interfaces (see "Interface Group Settings for Monitoring (NNM iSPI Performance for Metrics)" on page 374):

Prerequisite for these Interface Group thresholds:

1. Navigate to the **Interface Group Settings** form.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Expand the **Monitoring** folder.
   c. Select the **Monitoring Configuration** form.
   d. Locate the **Interface Group Settings** tab.
   e. Open the appropriate **Interface Group Settings** form.
2. Scroll down to the **SNMP Performance Monitoring** section, locate the Component Performance Monitoring settings. The ☑ Enable Interface Performance Polling must be enabled.

Available Interface Group Threshold Attributes for Monitoring Configuration

Relevant for:					
NNM iSPI Performance for Metrics Monitored Attributes available for Thresholds	**Interface Group: Threshold Settings**	**Interface Group: Baseline Settings**	**Node Group: Threshold Settings**	**Node Group: Baseline Settings**	
Order applied (low to high #):	**1st Group Order #s 1 - x**	**2nd Group Order #s 1 - x**	**3rd Group Order #s 1 - x**	**4th Group Order #s 1 - x**	**Controlled by:**
FCS LAN Error Rate *Local Area Network interfaces only*.Threshold based on the percentage of incoming frames with a bad checksum (CRC value) compared to the total number of incoming frames. Possible causes include collisions at half-duplex, a duplex mismatch, bad hardware (NIC, cable, or port), or a connected device generating frames with bad Frame Check Sequence.	X		X		Performance Polling
FCS WLAN Error Rate *Wireless Local Area Network Interfaces only*. Threshold based on the percentage of incoming frames with a bad checksum (CRC value) compared to the total number of incoming frames. Possible causes include wireless communication interference, bad hardware (NIC, cable or port), or a connected device generating frames with bad Frame Check Sequence.	X		X		Performance Polling
Input Discard Rate Threshold based on the percentage of the interface's discarded input packet	X		X		Performance Polling

Available Interface Group Threshold Attributes for Monitoring Configuration, continued

	Relevant for:				
NNM iSPI Performance for Metrics Monitored Attributes available for Thresholds	**Interface Group: Threshold Settings**	**Interface Group: Baseline Settings**	**Node Group: Threshold Settings**	**Node Group: Baseline Settings**	
Order applied (low to high #):	**1st Group Order #s 1 - x**	**2nd Group Order #s 1 - x**	**3rd Group Order #s 1 - x**	**4th Group Order #s 1 - x**	**Controlled by:**
count compared to the total number of packets received. Packets might be discarded because of a variety of issues, including receive-buffer overflows, congestion, or system specific issues.					
Input Error Rate Threshold based on the percentage of the interface's input packet error count compared to the total number of packets received. What constitutes an error is system specific, but likely includes such issues as bad packet checksums, incorrect header information, and packets that are too small.	X		X		Performance Polling
Input Queue Drops Rate Threshold based on the percentage of the interface's dropped input packets compared to the total number of packets received. Possible causes include the input queue being full.	X		X		Performance Polling
Input Utilization Threshold based on the percentage of the interface's total incoming octets compared to the maximum	X	X	X	X	Performance Polling

Available Interface Group Threshold Attributes for Monitoring Configuration, continued

	Relevant for:				
NNM iSPI Performance for Metrics Monitored Attributes available for Thresholds	**Interface Group: Threshold Settings**	**Interface Group: Baseline Settings**	**Node Group: Threshold Settings**	**Node Group: Baseline Settings**	
Order applied (low to high #):	**1st Group Order #s 1 - x**	**2nd Group Order #s 1 - x**	**3rd Group Order #s 1 - x**	**4th Group Order #s 1 - x**	**Controlled by:**
number of octets possible (determined by the MIB being used to query `ifSpeed` of the device and whether the host system supports high-speed counters for interfaces). **Tip:** Sometimes the `ifSpeed` value returned by the device's SNMP agent is not accurate and causes problems with thresholds. If your NNMi role allows, you can override the `ifSpeed` reported by the SNMP agent: 1. Open the problem interface's Interface form. 2. Select the General Tab. 3. Locate the Input/Output Speed section. 4. Change the Input Speed or Output Speed setting.					
Output Discard Rate Threshold based on the percentage of the interface's	X		X		Performance Polling

Available Interface Group Threshold Attributes for Monitoring Configuration, continued

	Relevant for:				
NNM iSPI Performance for Metrics Monitored Attributes available for Thresholds	**Interface Group: Threshold Settings**	**Interface Group: Baseline Settings**	**Node Group: Threshold Settings**	**Node Group: Baseline Settings**	
Order applied (low to high #):	**1st Group Order #s 1 - x**	**2nd Group Order #s 1 - x**	**3rd Group Order #s 1 - x**	**4th Group Order #s 1 - x**	**Controlled by:**
discarded output packet count compared to the total number of outgoing packets. Packets might be discarded because of a variety of issues, including transmission buffer overflows, congestion, or system specific issues.					
Output Error Rate Threshold based on the percentage of the interface's output packet error count compared to the total number of outgoing packets. What constitutes an error is system specific, but likely includes such issues as as collisions and buffer errors.	X		X		Performance Polling
Output Queue Drops Rate Threshold based on the percentage of the interface's dropped output packets compared to the total number of outgoing packets. Possible causes include all buffers allocated to the interface being full.	X		X		Performance Polling
Output Utilization Threshold based on the percentage of the interface's total outgoing octets compared to the maximum number of octets possible	X	X	X	X	Performance Polling

Available Interface Group Threshold Attributes for Monitoring Configuration, continued

	Relevant for:				
NNM iSPI Performance for Metrics Monitored Attributes available for Thresholds	Interface Group: Threshold Settings	Interface Group: Baseline Settings	Node Group: Threshold Settings	Node Group: Baseline Settings	
Order applied (low to high #):	1st Group Order #s 1 - x	2nd Group Order #s 1 - x	3rd Group Order #s 1 - x	4th Group Order #s 1 - x	Controlled by:
(determined by the MIB being used to query `ifSpeed` of the device and whether the host system supports high-speed counters for interfaces). **Tip:** Sometimes the `ifSpeed` value returned by the device's SNMP agent is not accurate and causes problems with thresholds. If your NNMi role allows, you can override the `ifSpeed` reported by the SNMP agent: 1. Open the problem interface's Interface form. 2. Select the General Tab. 3. Locate the Input/Output Speed section. 4. Change the Input Speed or Output Speed setting.					

Examples of Count-Based Threshold Monitoring

You can configure these example interface and node component thresholds if the HP Network Node Manager iSPI Performance for Metrics Software is installed. See "Purchase an HP Network

Node Manager i Smart Plug-in" on page 1486 for more information.

Several examples of Count-Based Threshold Settings are presented. These examples are not intended to be recommendations. Consider all aspects of your network environment and set performance thresholds that are meaningful in your environment:

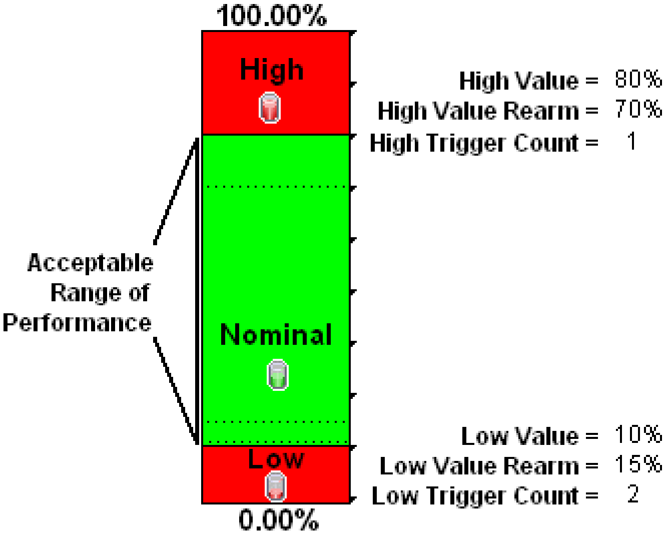
Example 1: Monitor Utilization on WAN Connections

You want to monitor the connections between two sites to verify that your service provider is meeting their guaranteed throughput volume. You pay a fixed cost for a specific bandwidth over this WAN interface.

- Monitor for under-utilization which wastes money (less than 10%).

> **Tip:** If you do not care about under-utilization, set Low Value and Low Value Rearm to 0% as shown in Example 2. The Low Value threshold is then disabled because it cannot be *crossed*.

- Monitor for over-utilization (greater than 80%), which might result in performance bottlenecks or service provider surcharges.

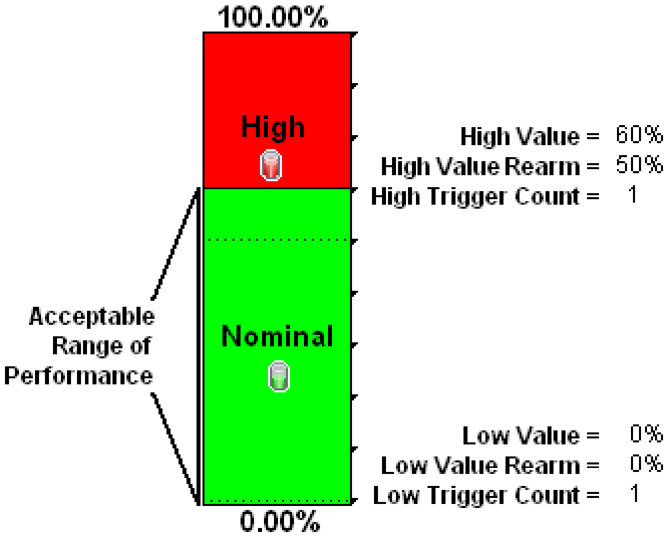


> **Note:** Sometimes an Interface's MIB-II ifspeed value is not reported accurately.
>
> This might result in threshold calculations outside the 0.00 - 100.00 range. If this happens, the Interface threshold State set to "Unavailable." To correct the problem:
>
> 1. Access the **Inventory** workspace
> 2. Open **Interface** view.
> 3. Open the form for the Interface that is reporting a threshold state of "Unavailable."
> 4. Navigate to the **General** tab.
> 5. Enter a valid entry in **Input Speed** or **Output Speed** (this overrides the value returned by the device's SNMP agent so that NNMi can accurately calculate utilization thresholds).

Example 2: Monitor Utilization on Important Interfaces

You want to monitor an important Ethernet interface and be notified if it is getting overloaded.

An Ethernet interface configured for full-duplex operation has an acceptable operating range of 0-60%. When average utilization is greater than 60%, you want NNM to generate a High Threshold incident.



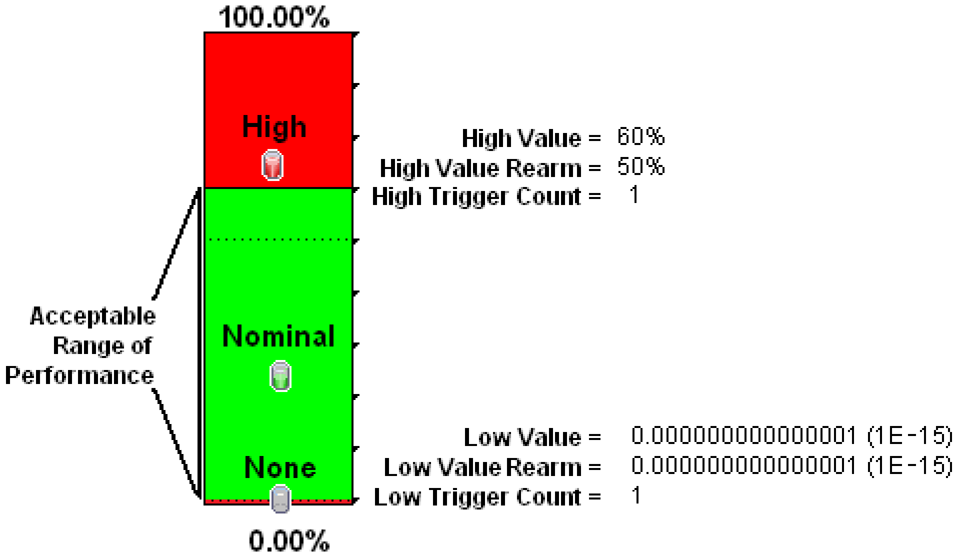
Example 3: Monitor Utilization on Important Interfaces for States (High, Nominal, None)

You want to monitor an important Ethernet interface and be notified if it is getting overloaded or if no data has passed through the interface during the polling interval. This might indicate a problem with the interface or its connection. If a formerly connected interface is *Administratively Down*, NNMi honors that and does not generate a fault condition.

This example monitors for the following:

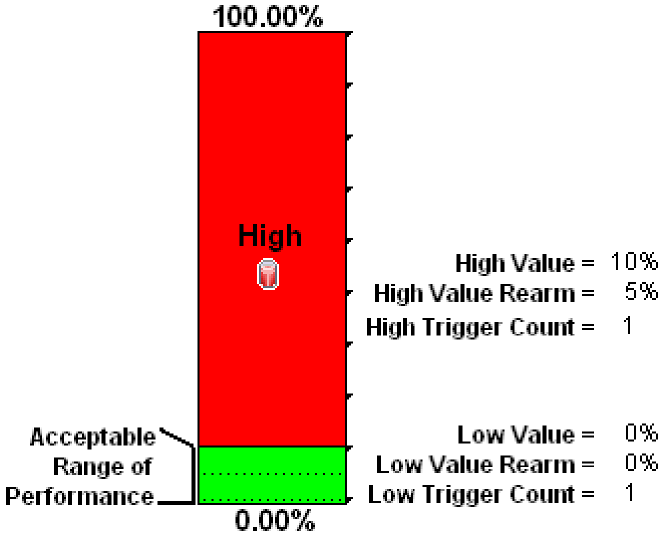
- When the average utilization is greater than 60%
- When zero data is passing through the interface

> **Tip:** The Low Value of 0.000000000000001 used in this example because it is the smallest value greater than zero available in NNMi. When you configure a Threshold, to use this value, simply type `1E-15` and press Enter. NNMi converts that Scientific Notation to the text string 0.000000000000001 (with 1 entered in the 15th position after the decimal).


100.00%
High
High Value = 60%
High Value Rearm = 50%
High Trigger Count = 1
Acceptable Range of Performance
Nominal
Low Value = 0.000000000000001 (1E-15)
Low Value Rearm = 0.000000000000001 (1E-15)
Low Trigger Count = 1
None
0.00%

Example 4: Monitor Important Interfaces for Discards

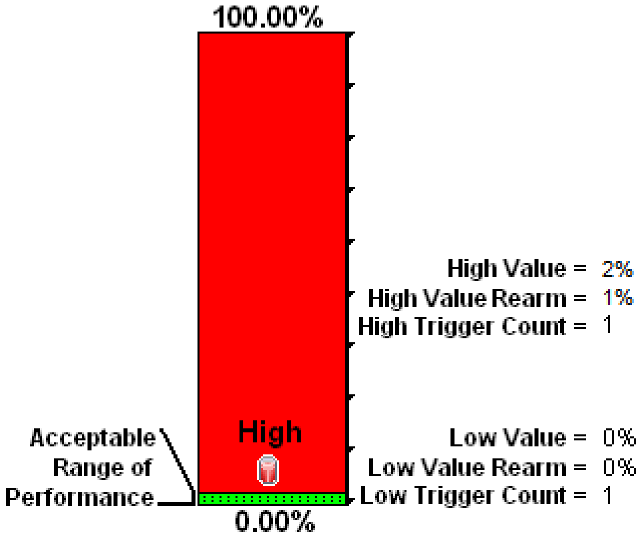
You want to know any time an interface is dropping data. The acceptable limit for interface discards is 10%. A High Threshold situation occurs when the discard rate exceeds 10% and returns to Nominal when the discard rate drops below 5%.



Example 5: Monitor Important Interfaces for Errors

You want to know if packet errors occur. The acceptable limit for packet errors is 2%.A High Threshold situation occurs when the error rate exceeds 2% and returns to normal when the error rate drops below 1%.

Tip: To monitor for any errors greater than zero, set the **High Value**, **High Value Rearm**, **Low Value**, and **Low Value Rearm** to: 0.0



Examples of Time-Based Threshold Monitoring

You can configure these example interface and node component thresholds if the HP Network Node Manager iSPI Performance for Metrics Software is installed. See "Purchase an HP Network Node Manager i Smart Plug-in" on page 1486 for more information.

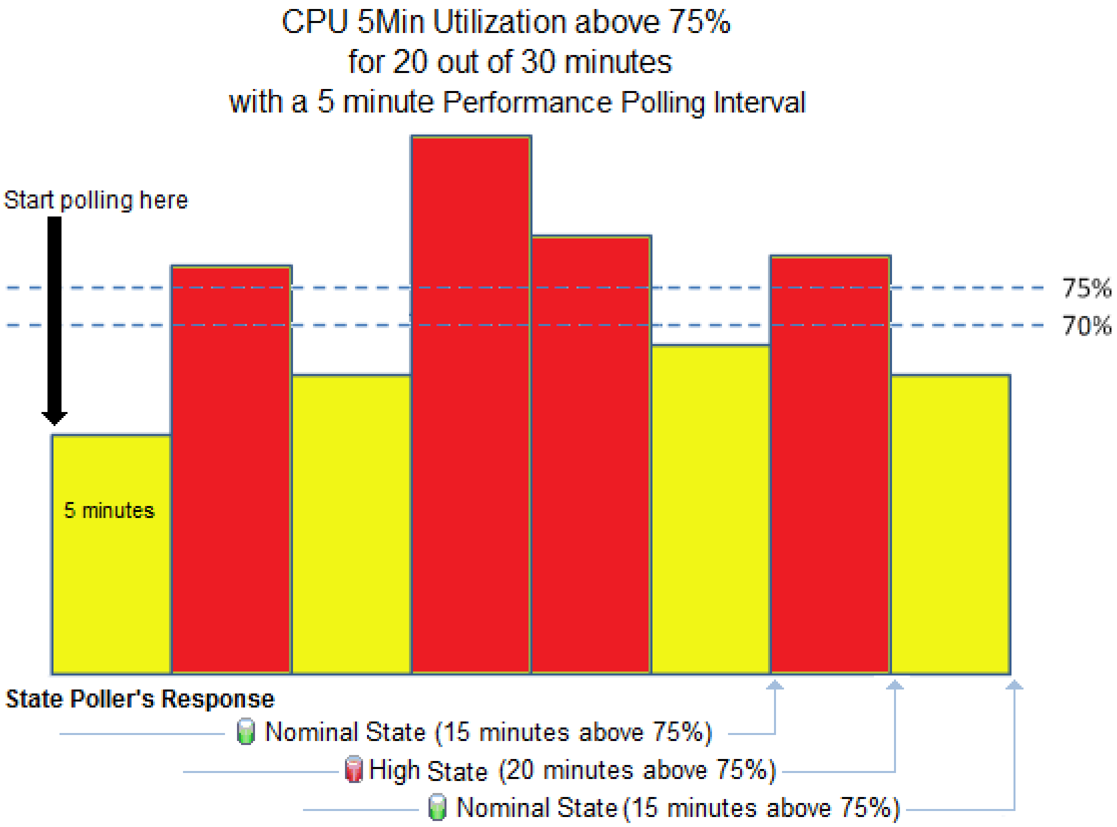
Several Time-Based Threshold Settings examples are presented. These examples are not intended to be recommendations. Consider all aspects of your network environment and set performance thresholds that are meaningful in your environment:

Example 1: Monitor CPU Utilization for an Important Node

You want to know when the CPU Utilization is above 75% for 20 out of 30 minutes. A High Threshold situation occurs when the CPU Utilization exceeds 75% for 20 out of 30 minutes and returns to normal when the Utilization drops below 70%.

High Value: 75%
High Value Rearm: 70%
High Duration: 20 minutes
High Duration Window: 30 minutes

Note: The influencing Performance Polling Interval is set to 5 minutes, which instructs NNMi to poll the value every 5 minutes.

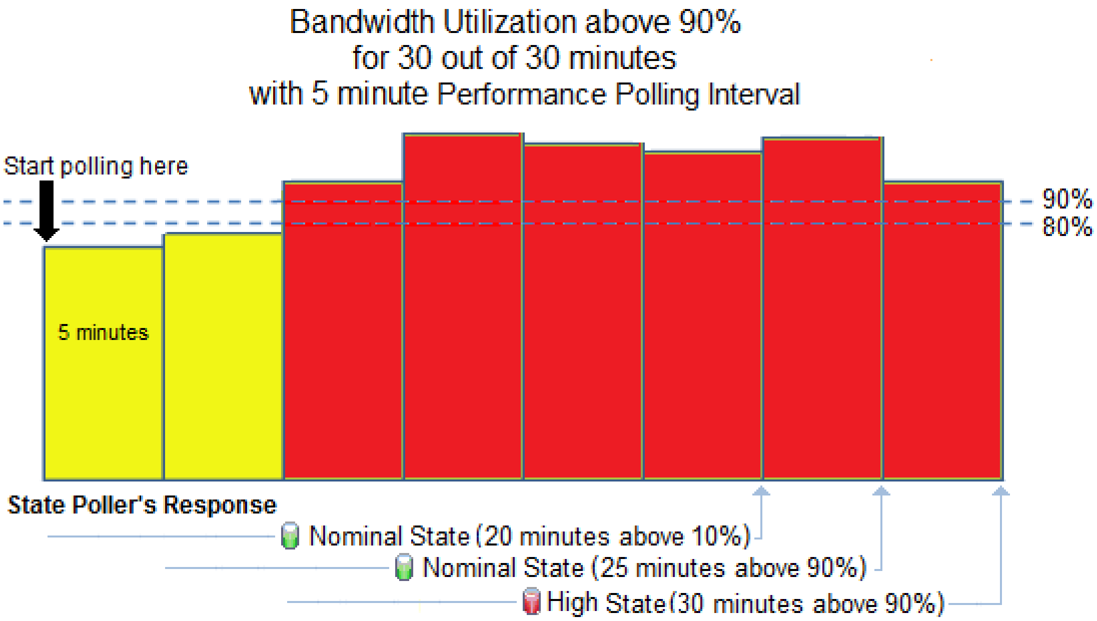


Example 2: Monitor Important Interfaces for Interface Input Utilization

You want to know when the Bandwidth Utilization is above 90% for 30 out of 30 minutes. A High Threshold situation occurs when the bandwidth utilization exceeds 90% for 30 out of 30 minutes and returns to normal when the utilization drops below 80%.

High Value: 90%
High Value Rearm: 80%
High Duration: 30 minutes
High Duration Window: 30 minutes

Note: The influencing Performance Polling Interval is set to 5 minutes, which instructs NNMi to poll the value every 5 minutes.

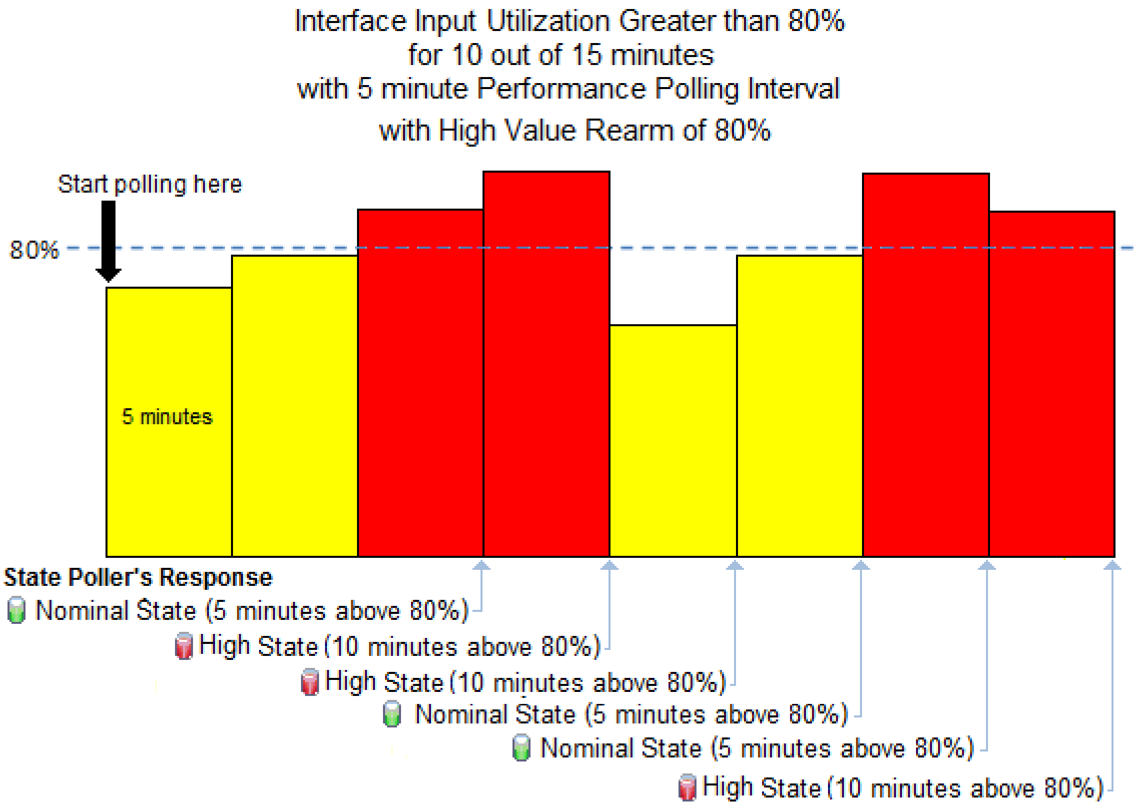


Example 3: Monitor Using Rearm Values

You want to reduce the frequency of interface State changes when the polled value is close to the threshold.

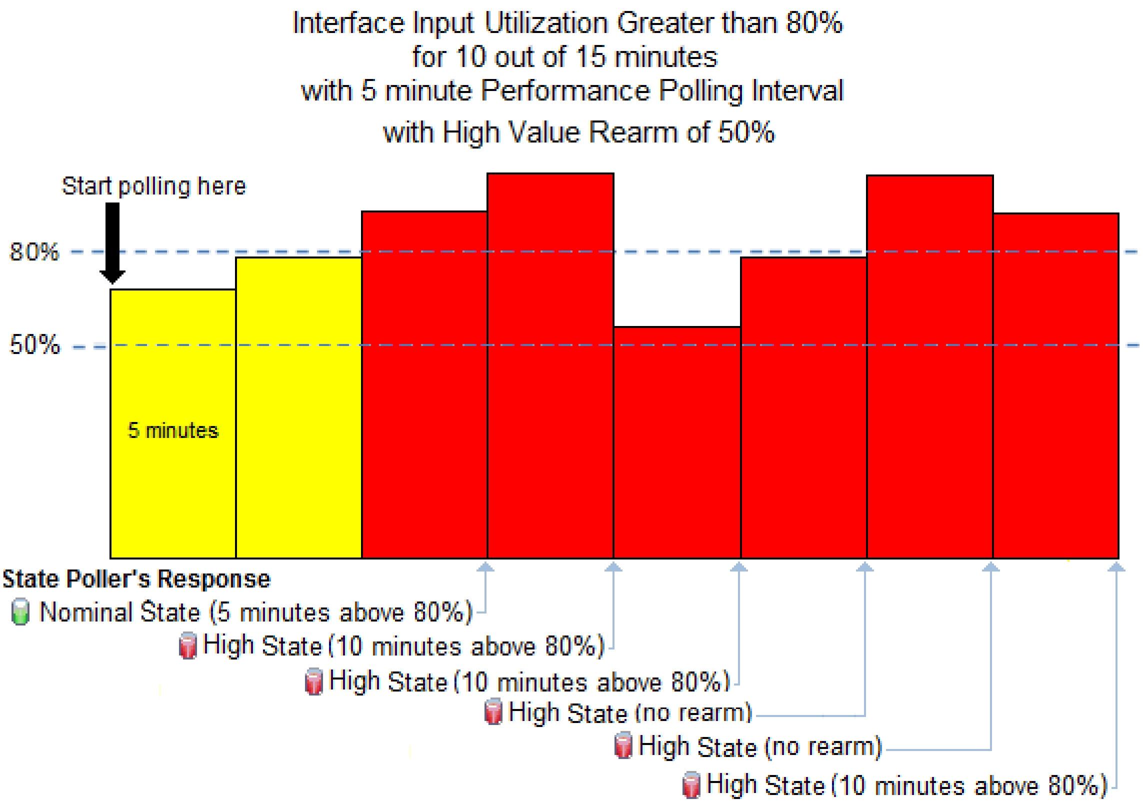
The first example shows the Performance Polling results with **High Value Rearm** set to 80% (same percentage as the threshold).

High Value: 80%
High Value Rearm: 80%
High Duration: 10 minutes
High Duration Window: 15 minutes



The second example shows the same set of Performance Polling results with **High Value Rearm** set to 50%.

High Value: 80%
High Value Rearm: 50%
High Duration: 10 minutes
High Duration Window: 15 minutes

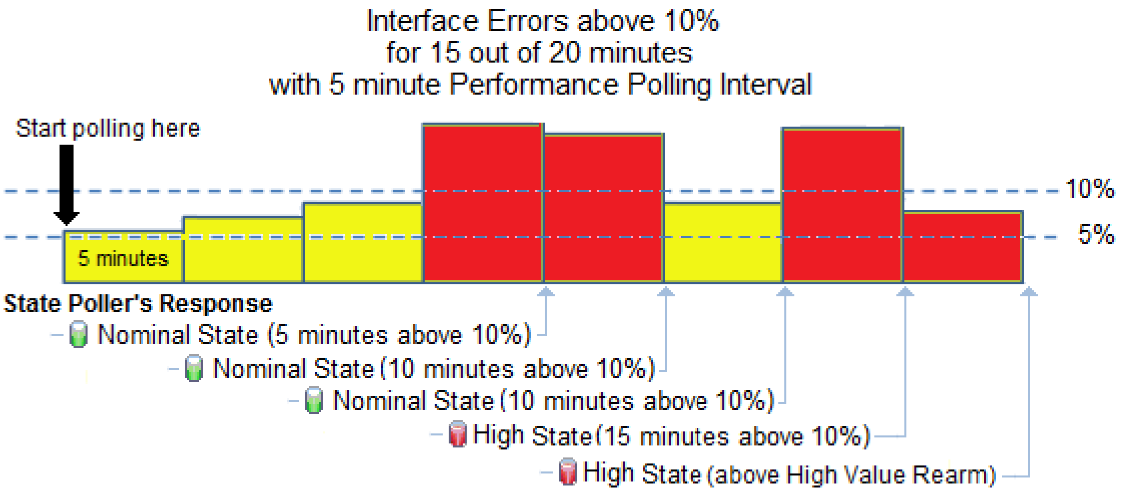


Example 4: Monitor Important Interfaces for Interface Errors

You want to know when the Interface Errors are above 10% for 15 out of 20 minutes. A High Threshold situation occurs when the interface errors exceed 10% for 15 out of 20 minutes and returns to normal when the interface errors drops below 5%.

High Value: 10%
High Value Rearm: 5%
High Duration: 15 minutes
High Duration Window: 20 minutes

Note: The influencing Performance Polling Interval is set to 5 minutes, which instructs NNMi to poll the value every 5 minutes.

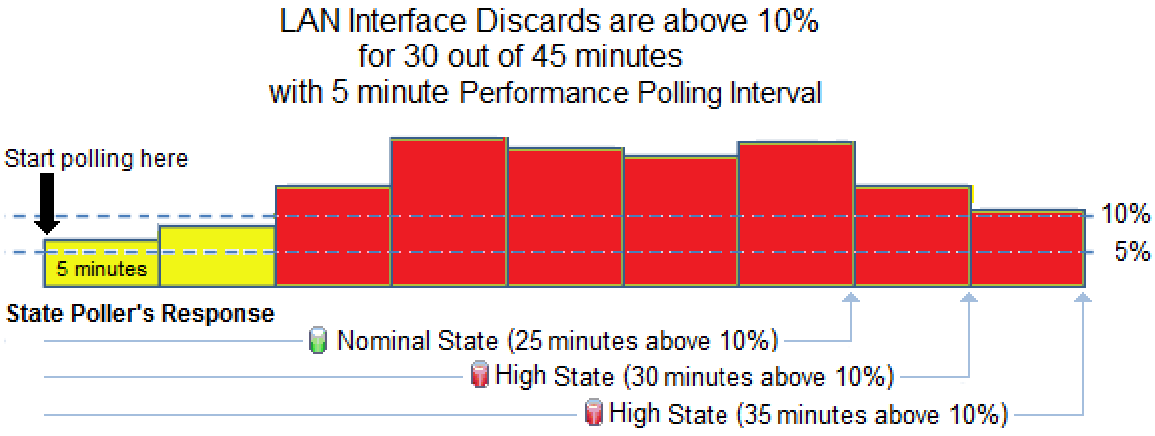


Example 5: Monitor Important Interfaces for Interface Discards

You want to know when the Interface Discards are above 10% for 30 out of 45 minutes. A High Threshold situation occurs when the interface discards exceed 10% for 30 out of 45 minutes and returns to nomral when the interface discards drop below 5%.

High Value: 10%
High Value Rearm: 5 %
High Duration: 30 minutes
High Duration Window: 45 minutes

> **Note:** The influencing Performance Polling Interval is set to 5 minutes, which instructs NNMi to poll the value every 5 minutes.



Interface Group Settings for Monitoring (*NNM iSPI Performance for Metrics*)

Before you start, you must establish one or more Interface Group definitions that identify the interface types to which these monitoring settings will apply. NNMi provides nearly 250 interface types to choose from. Interface monitoring applies to matching interfaces and the IP addresses that are hosted on those interfaces. See also, "Interface Groups Provided by NNMi" on page 338.

> **Tip:** NNMi administrators can check network latency for an Interface Group by adjusting the following for the management addresses associated with the specified group of interfaces:
>
> - ICMP polling frequency
> - ICMP echo request packet data payload size
>
> See the "Maintaining NNMi" chapter in the *HP Network Node Manager i Software Deployment Reference* for more information.

> **Tip:** (*NNMi Advanced*) Global Network Management feature - When viewing maps on the Global Manager, if you want to monitor important WAN interface connections *between Regional Managers*, then within each Regional Manager's Monitoring Configuration settings, enable NNMi's Poll Unconnected Interfaces for each of those WAN interfaces.

To establish monitoring behavior for one or more predefined Interface Groups:

1. Navigate to the **Interface Group Settings** form.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Expand the **Monitoring** folder.
   c. Select **Monitoring Configuration**.
   d. Locate the **Interface Group Settings** tab.
   e. Do one of the following:
      - To create an Interface Group Settings definition, click the New icon.
      - To edit an Interface Group Settings definition, select a row and click the Open icon.
      - To delete an Interface Group Settings definition, select a row and click the Delete button
2. Establish the appropriate settings to identify this Interface Group Setting definition (see Basics table).
3. *Optional*. Configure the Fault Monitoring behavior for this Interface Group Setting definition (see Fault Monitoring table).
4. (*NNM iSPI Performance for Metrics*) If the HP Network Node Manager iSPI Performance for Metrics Software is installed:

- Configure the Performance Monitoring behavior for this Interface Group Setting definition. See Performance Monitoring table.
- Configure the Baseline Settings. Navigate to the Baseline Settings tab. See "Configure Baseline Settings for Interface Groups" on page 389.

5. *Optional*. Configure thresholds. Navigate to the Threshold Settings tab. See "Configure Threshold Monitoring for Interface Groups" on page 381 for more information.
6. By default, NNMi monitors only interfaces that are connected to other interfaces. When SNMP polling is enabled, NNMi automatically detects most connections. See "Add or Delete a Layer 2 Connection" on page 284 for information about manual overrides.

   Optional. If you want to expand monitoring behavior for this group to include unconnected Interfaces, indicate your choices in the Extend the Scope of Polling Beyond Connected Interfaces group box.
7. Click **Save and Close** to return to the Monitoring Configuration form.
8. Click **Save and Close**. NNMi applies your changes. The next regularly scheduled monitoring cycle uses the new settings.

   > **Caution:** When you establish monitoring configuration settings, NNMi must recalculate the settings for all affected objects. This can take some time and slow down your system. Consider making this change during a slow time in your network environment.

   To verify that State Poller is working as expected, see **Help** → **System Information** and select the the **State Poller** tab. NNMi displays a report with current details about the State Poller process.

Optional. Customize the node monitoring behavior. See "Node Group Settings for Monitoring" on page 391. Also see "Detect Interface Changes" on page 280.

Basics

Attribute	Description
Ordering	Enter a unique string (any length), characters 0 through 9. Consider using increments of 100 for the flexibility to insert additional items between existing items over time. NNMi decides which monitoring configurations apply to a node or interface based on the ordering number assigned to the configuration definitions. NNMi monitors the device according to the first match (checked from lowest number to highest number within each category). Categories are read in sequence. Click here for a description of the sequence. 1. **Interface Group Settings**: NNMi monitors each of the Node's Interfaces and IP Addresses based on the first matching Interface Group Settings definition. The first match is the Interface Group Settings definition with the lowest Ordering number, then Baseline Settings. 2. **Node Group Settings**: NNMi monitors each Node and each previously unmatched Interface or IP Address based on the first matching Node Group Settings definition. The first match is the Node Group Settings definition with the lowest Ordering number, then Baseline Settings.

Basics, continued

Attribute	Description
	Note: Child node groups are included in the Ordering hierarchy. This means that if the parent node group has a lower Ordering number (for example, parent=10, child=20), then the monitoring configuration specified for the parent node group also applies to the nodes in the child node group. To override a parent node group monitoring configuration, set the Ordering number for the child node group to a number that is lower than the parent (for example, parent=20, child=10). 3. **Default Settings**: If no match is found for a Node, Interface, or IP Address in 1 or 2, NNMi applies the default Monitoring Configuration settings. No duplicate Ordering numbers are permitted. Each Interface Setting ordering number must be unique.
Interface Group	Choose one predefined Interface Group from the list. See "Create Interface Groups" on page 321 for more information. (*NNMi Advanced with IPv6 enabled*) See also "Interface Groups of IPv4 or IPv6 Addresses" on page 331.

Fault Monitoring

Attribute	Description
Enable ICMP Monitoring: Enable IP Address Fault Polling **Note:** This monitoring option is useful for devices that do not	If ☑ enabled, State Poller issues ICMP (ping) requests to verify the availability of discovered IP address. **Note:** In the Global Control section of this form, the Enable State Polling attribute must be enabled, too. If □ disabled, State Poller does the following: • If neither this attribute nor *Management IP Address Polling* is selected, State Poller does not use ICMP to monitor nodes covered by this configuration setting. • IP addresses (both previously discovered and newly discovered) have a State attribute value of "Not Polled" and a Status attribute value of "No Status" with the color of the IP address map-symbol set to beige. See Layer 3 Neighbor View. • If both ICMP and SNMP are disabled for a Node, the Node has a Status attribute value of "No Status" and the color of the Node map-symbol background shape is set to beige.

Fault Monitoring , continued

Attribute	Description
support SNMP.	**Tip:** To turn off ICMP polling within a subset of your network environment, use the Communication Configuration workspace Region definitions. You can define your own Regions that identify any unreachable addresses in your management domain (for example, the **private IP addresses**[1]).
SNMP Fault Monitoring: Enable Interface Fault Polling	If ☑ enabled, State Poller monitors all interfaces by issuing SNMP read-only queries to devices assigned to this level of the monitoring hierarchy. By default, any connected interface is monitored for MIB-II ifAdminStatus and ifOperStatus. (ifAdminStatus is set by the device administrator. ifOperStatus indicates the operational status of interface health.) If you have *unconnected* interfaces that you want to monitor, expand NNMi monitoring behavior with the Poll Unconnected Interfaces and the Poll Interfaces Hosting IP Addresses attributes. **Note:** The following attributes must also be enabled: • In the Global Control section of this form, the Enable State Polling attribute must be enabled, too. See Layer 2 Neighbor View. (See "Global Control Settings for Monitoring" on page 343 for more information.) • In the Communication Configuration view, enable State Poller queries with the applicable Enable SNMP Communication attributes (see "Configuring Communication Protocol" on page 119 for more information). If □ disabled, for devices assigned to this level of the monitoring hierarchy: • Causal Engine calculates Status based only on IP address State. • The Interface objects previously discovered change to a State attribute value of "Not Polled" and a Status attribute value of "No Status" (plus any related map-symbol changes to a beige color).
Fault Polling Interval	The time that State Poller waits between issuing queries to gather information for any of the following that are enabled: ICMP Polling, SNMP Polling, Poll Unconnected Interfaces, and Poll Interfaces Hosting IP addresses. The default Fault Polling Interval is 5 minutes, except for the Node Group named Microsoft Windows Systems which is 10 minutes. **Note:** NNMi monitors SNMP agents (Management Addresses) according to this Fault Polling Interval, *even if ICMP Polling, SNMP Polling, Poll Unconnected Interfaces, and Poll Interfaces Hosting IP addresses are all*

[1] These are IPv4 addresses that can be reused in home and office local area networks (LANs). Following the standards set by RFC 1918 and RFC 4193 (10.*.*.*, 169.254.*.*, 172.16-31.*.*, and 192.168.*.*)

Fault Monitoring , continued

Attribute	Description
	disabled. To prevent an SNMP Agent's address from being monitored, one of the following must be true: State Polling is disabled, current Communication Configuration settings turn off SNMP for the SNMP agent's address, the parent Node is set to Not Managed or Out of Service, or the parent node belongs to a Monitoring Configuration's Node Group with □ **Enable SNMP Polling on Node** disabled.

SNMP Performance Monitoring (*NNM iSPI Performance for Metrics*)

Attribute	Description
LAN Performance Monitoring: Enable Interface Performance Polling	(*NNM iSPI Performance for Metrics*) Use this attribute to extend the range of polling data that NNMi collects. HP Network Node Manager iSPI Performance for Metrics Software uses the additional data in a series of performance reports. See "Purchase an HP Network Node Manager i Smart Plug-in" on page 1486 for more information. When enabled, network traffic increases on your network because NNMi gathers performance data about each member of this group on a regular schedule. If ☑ enabled, NNMi gathers basic Interface performance data from Interfaces in devices assigned to this level of the monitoring hierarchy. If □ disabled, NNMi does not extend data collection behavior to include performance data about Interfaces assigned to this level of the monitoring hierarchy. **Note:** The Enable State Polling field must be enabled, too. By default the performance of connected interfaces and addresses is monitored. If you have *unconnected* interfaces that you want to monitor, expand NNMi monitoring behavior by enabling Poll Unconnected Interfaces.
WAN Performance Monitoring: Enable DSx Interface Performance Polling	(*NNM iSPI Performance for Metrics*) Use this attribute to extend the range of polling data that NNMi collects. HP Network Node Manager iSPI Performance for Metrics Software uses the additional data in a series of performance reports. When enabled, network traffic increases on your network because NNMi gathers performance data about each member of the **DSx Interfaces** interface group on a regular schedule. See "Interface Groups Provided by NNMi" on page 338 for more information. If ☑ enabled, NNMi gathers DSx performance data from DSx Interfaces assigned to this level of the monitoring hierarchy. If □ disabled, NNMi does not gather DSx performance data from DSx Interfaces assigned to this level of the monitoring hierarchy.
WAN Performance Monitoring:	(*NNM iSPI Performance for Metrics*) Use this attribute to extend the range of polling data that NNMi collects. HP Network Node Manager iSPI

SNMP Performance Monitoring (NNM iSPI Performance for Metrics), continued

Attribute	Description
Enable SONET Interface Performance Polling	Performance for Metrics Software uses the additional data in a series of performance reports. When enabled, network traffic increases on your network because NNMi gathers performance data about each member of the **SONET Interfaces** interface group on a regular schedule. See "Interface Groups Provided by NNMi" on page 338 for more information. If ☑ enabled, NNMi gathers SONET performance data from SONET Interfaces assigned to this level of the monitoring hierarchy. If □ disabled, NNMi does not gather SONET performance data from SONET Interfaces assigned to this level of the monitoring hierarchy.
WAN Performance Monitoring: Enable ATM Interface Performance Polling	(*NNM iSPI Performance for Metrics*) Use this attribute to extend the range of polling data that NNMi collects. HP Network Node Manager iSPI Performance for Metrics Software uses the additional data in a series of performance reports. When enabled, network traffic increases on your network because NNMi gathers performance data for each ATM Interface. If ☑ enabled, NNMi gathers ATM performance data from ATM Interfaces assigned to this level of the monitoring hierarchy. If □ disabled, NNMi does not gather ATM performance data from ATM Interfaces assigned to this level of the monitoring hierarchy. **Note:** • This option gathers metrics from ATM-MIB and CISCO-AAL5-MIB. • See also "Configure Discovery of ATM/Frame Relay Interfaces" on page 201.
WAN Performance Monitoring: Enable Frame Relay Interface Performance Polling	(*NNM iSPI Performance for Metrics*) Use this attribute to extend the range of polling data that NNMi collects. HP Network Node Manager iSPI Performance for Metrics Software uses the additional data in a series of performance reports. When enabled, network traffic increases on your network because NNMi gathers performance data for each Frame Relay Interface. If ☑ enabled, NNMi gathers Frame Relay performance data from Frame Relay Interfaces assigned to this level of the monitoring hierarchy. If □ disabled, NNMi does not gather Frame Relay performance data from Frame Relay Interfaces assigned to this level of the monitoring hierarchy. This option gathers the following types of metrics: • Circuit in and out octets, errors, and discards • Committed Information Rate (CIR) and Extended Information Rate (EIR) utilization

SNMP Performance Monitoring (NNM iSPI Performance for Metrics), continued

Attribute	Description
	• Forward Error Congestion Notification (FECN) and Backward Error Congestion Notification (BECN) counts See also "Configure Discovery of ATM/Frame Relay Interfaces" on page 201.
Performance Polling Interval	(*NNM iSPI Performance for Metrics*) Use this field to set the time period that NNMi waits between issuing network traffic to gather performance data for the HP Network Node Manager iSPI Performance for Metrics Software. The default Performance Polling Interval is 5 minutes, except for the Node Group named Microsoft Windows Systems which is 10 minutes.

Extend the Scope of Polling Beyond Connected Interfaces

Attribute	Description
Poll Unconnected Interfaces	If ☑ enabled, NNMi monitors all interfaces within discovered devices (both connected and unconnected). All interfaces are monitored for MIB-II ifAdminStatus and ifOperStatus. (ifAdminStatus is set by the device administrator. ifOperStatus indicates the operational status of interface health.) **Note:** The Enable State Polling field must be enabled, and SNMP polling of some type must be enabled (for example, Enable SNMP Fault Monitoring and Enable SNMP Performance Polling). If □ disabled, State Poller polls according to other configuration settings. **Tip:** Your discovery configuration choices might need to be adjusted to get the results you want. For example, to meet the “connected” criteria for interfaces in switches that do not have an IP address you must add the device to which the interface is connected as a discovery seed. See"Specify Discovery Seeds" on page 256.
Poll Interfaces Hosting IP Addresses **Note:** This monitoring option is useful for	If ☑ enabled, any unconnected interface that has one or more addresses associated with it is monitored for MIB-II ifAdminStatus and ifOperStatus. (ifAdminStatus is set by the device administrator. ifOperStatus indicates the operational status of interface health.) **Note:** The Enable State Polling field must be enabled, and SNMP polling of some type must be enabled (for example, Enable SNMP Fault Monitoring and Enable SNMP Performance Polling). By monitoring the Interface (in addition to the IP address), NNMi can make more informed decisions about the health of each IP address associated with an unconnected interface.

Extend the Scope of Polling Beyond Connected Interfaces, continued

Attribute	Description
Router interfaces.	If □ disabled, State Poller polls according to other configuration settings. **Tip:** The Communication Configuration workspace provides a method of overriding this setting for specific Regions. You can define your own Region to easily turn off polling to any unreachable addresses in your management domain (for example, the **private IP addresses**[1]).

Related Topics

"Threshold Monitoring Behavior After a System Restart or Configuration Change" on page 415

Configure Threshold Monitoring for Interface Groups

You can set interface thresholds using either of the following methods:

- "Configure Count-Based Threshold Monitoring for Interface Groups" below
- "Configure Time-Based Threshold Monitoring for Interface Groups" on page 385

Related Topics

"About Threshold Settings Provided by NNMi" on page 354

"Threshold Monitoring Behavior After a System Restart or Configuration Change" on page 415

Configure Count-Based Threshold Monitoring for Interface Groups

Count-Based Threshold Settings enable you to determine as soon as a threshold is reached (for example, an interface is dropping data or an Ethernet interface is getting overloaded).

To establish count-based threshold monitoring behavior for interfaces:

1. *Prerequisite*. Before setting thresholds, analyze performance data over time to determine wise threshold settings for each Interface group. For more information, see the following topics:
   - "Determine Reasonable Threshold Settings" on page 413.
   - "Examples of Count-Based Threshold Monitoring" on page 364.
2. Navigate to the **Interface Group Settings** form.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Expand the **Monitoring** folder.
   c. Select **Monitoring Configuration**.

[1] These are IPv4 addresses that can be reused in home and office local area networks (LANs). Following the standards set by RFC 1918 and RFC 4193 (10.*.*.*, 169.254.*.*, 172.16-31.*.*, and 192.168.*.*)

   d. Navigate to the **Interface Group Settings** tab.

   e. Do one of the following:

      ○ To create an Interface Group Settings definition, click the New icon.

      ○ To edit an Interface Group Settings definition, select a row and click the Open icon.

3. In the **Interface Group Settings** form, navigate to the **Threshold Settings** tab.

4. Do one of the following:

   - To create a threshold definition, click the New icon and select **Count-Based Threshold Settings**.

   - To edit a threshold definition, select a row and click the Open icon.

   - To delete a threshold definition, select a row and click the Delete icon.

5. Select the Monitored Attribute you want to monitor and establish the threshold values for that attribute (see Basic Count-Based Threshold Settings table).

   When you configure thresholds using this technique, NNMi uses the assigned Interface Group as a filter (only monitoring the threshold for devices with at least one interface belonging to the specified Interface Group).

6. Click **Save and Close** to return to the **Interface Group Settings** form.

7. Click **Save and Close** to return to the **Monitoring Configuration** form.

8. Click **Save and Close**. NNMi applies your changes during the next regularly scheduled monitoring cycle.

   > **Note:** Threshold Incidents are disabled by default within NNMi to prevent Incident storms. If you are ready to generate Threshold Incidents, see "Generate Performance Threshold Incidents (NNM iSPI Performance for Metrics)" on page 764. See also "Custom Incident Attributes Provided by NNMi (Information for Administrators)" on page 647 for a description of the special custom incident attributes available in Threshold Incidents.

9. See also "Find Threshold Results" on page 414.

Basic Count-Based Threshold Settings

Attribute	Description
Monitored Attribute	In the Monitored Attribute drop-down list, select the attribute for which you want to establish a threshold configuration. **Tip:** Some of the choices in the Monitored Attribute selection list do not apply in this context. See the tables in "About Threshold Settings Provided by NNMi" on page 354 for information about which Monitored Attributes are available for Interface Groups.
A High Threshold situation occurs when:	

Basic Count-Based Threshold Settings, continued

Attribute	Description
	The *Monitored Attribute* is greater than the *High Value* for *High Trigger Count* cycles. When these criteria are met, NNMi does the following: - Updates the Threshold's state value to **High** for the appropriate Interface. - Generates the related incident (if one is Enabled ☑). If an incident is generated, NNMi closes that incident when the Threshold criteria are no longer met.
High Value	Designate the percentage between 0.00 and 100.00 above which becomes a threshold situation. For special situations, the following values can be used: - 0.000000000000001 (or 1E-15 in Scientific Notation) for the smallest value greater than zero. - 99.99999999999999 for the highest value less than one hundred. The High Value must be greater than or equal to the designated Low Value. **Note:** If you use the highest possible value, the High threshold is disabled because it cannot be *crossed*.
High Value Rearm	The High Value Rearm designates the lower boundary of the High Threshold *range of values*. After entering a High threshold situation, when a returned value is below the specified High Value Rearm, the High Threshold situation ends (for Count-Based Thresholds). **Note:** The High Value Rearm must be less than or equal to the High Value and greater than or equal to the Low Value Rearm.
High Trigger Count	Designate the number of consecutive polling intervals the returned value must be greater than the specified High Value to meet the High Threshold criteria. The default value is 1. The polled value represents an average over the configured polling interval, so a trigger count of 1 is often appropriate. See the currently configured *Fault Polling Interval* or *Performance Polling Interval* setting that is influencing the Monitored Attribute you chose, because that is how often NNMi provides a data point. See the tables in "About Threshold Settings Provided by NNMi" on page 354 for details. See the following topics for instructions about finding the current polling interval setting: - "Default Settings for Monitoring" on page 345 - "Interface Group Settings for Monitoring (NNM iSPI Performance for Metrics)" on page 374 - "Node Group Settings for Monitoring" on page 391

Basic Count-Based Threshold Settings, continued

Attribute	Description
A Low Threshold situation occurs when: The *Monitored Attribute* is less than the *Low Value* for *Low Trigger Count* cycles. When these criteria are met, NNMi does the following: • Updates the Threshold's state value to **Low** for the appropriate Interface. • Generates the related incident (if one is Enabled ☑). If an incident is generated, NNMi closes that incident when the Threshold criteria are no longer met.	
Low Value	Designate the percentage between 0.00 and 100.00 below which becomes a threshold situation. For special situations, the following values can be used: • 0.000000000000001 (or 1E-15 in Scientific Notation) for the smallest value greater than zero. • 99.99999999999999 for the highest value less than one hundred. The Low Value must be less than or equal to the designated High Value. **Note:** If you use zero (the minimum possible value), the Low threshold is disabled because it cannot be *crossed*.
Low Value Rearm	The Low Value Rearm designates the upper boundary of the Low Threshold *range of values*. After entering a Low threshold situation, when a returned value is above the specified Low Value Rearm, the Low Threshold situation ends (for Count-Based Thresholds). **Note:** The Low Value Rearm must be greater than or equal to the Low Value and less than or equal to the High Value Rearm.
Low Trigger Count	Designate the number of consecutive polling interval the returned value must be less than the specified Low Value to meet the Low Threshold criteria. The default value is 1. The polled value represents an average over the configured polling interval, so a trigger count of 1 is often appropriate. See the currently configured *Fault Polling Interval* or *Performance Polling Interval* setting that is influencing the Monitored Attribute you chose, because that is how often NNMi provides a data point. See the tables in "About Threshold Settings Provided by NNMi" on page 354 for details. See the following topics for instructions about finding the current polling interval setting: • "Default Settings for Monitoring" on page 345 • "Interface Group Settings for Monitoring (NNM iSPI Performance for Metrics)" on page 374 • "Node Group Settings for Monitoring" on page 391

Configure Time-Based Threshold Monitoring for Interface Groups

Time-Based Threshold Settings enable you to determine whether a threshold is reached for a particular duration of time (for example, the bandwidth utilization for an interface is above 90 percent for 20 out of 30 minutes).

To establish time-based threshold monitoring behavior for interfaces:

1. *Prerequisite*. Before setting thresholds, analyze performance data over time to determine wise threshold settings for each Interface group.
   - "Determine Reasonable Threshold Settings" on page 413.
   - "Examples of Time-Based Threshold Monitoring" on page 368 .
2. Navigate to the **Interface Group Settings** form.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Expand the **Monitoring** folder.
   c. Select **Monitoring Configuration**.
   d. Navigate to the **Interface Group Settings** tab.
   e. Do one of the following:
      - To create an Interface Group Settings definition, click the New icon.
      - To edit an Interface Group Settings definition, select a row and click the Open icon.
3. In the **Interface Group Settings** form, navigate to the **Threshold Settings** tab.
4. Do one of the following:
   - To create a threshold definition, click the New icon and select **Time-Based Threshold Settings**.
   - To edit a threshold definition, select a row and click the Open icon.
   - To delete a threshold definition, select a row and click the Delete icon.
5. Select the Monitored Attribute you want to monitor and establish the threshold values for that attribute (see Basic Time-Based Threshold Settings table).

   When you configure thresholds using this technique, NNMi uses the assigned Interface Group as a filter (only monitoring the threshold for devices with at least one interface belonging to the specified Interface Group).
6. Click **Save and Close** to return to the **Interface Group Settings** form.
7. Click **Save and Close** to return to the **Monitoring Configuration** form.
8. Click **Save and Close**. NNMi applies your changes during the next regularly scheduled monitoring cycle.

> **Note:** Threshold Incidents are disabled by default within NNMi to prevent Incident storms.

> If you are ready to generate Threshold Incidents, see "Generate Performance Threshold Incidents (NNM iSPI Performance for Metrics)" on page 764. See also "Custom Incident Attributes Provided by NNMi (Information for Administrators)" on page 647 for a description of the special custom incident attributes available in Threshold Incidents.

9. See also "Find Threshold Results" on page 414.

Basic Time-Based Threshold Settings

Attribute	Description
Monitored Attribute	In the Monitored Attribute drop-down list, select the attribute for which you want to establish a threshold configuration. **Tip:** Some of the choices in the Monitored Attribute selection list do not apply in this context. See the tables in "About Threshold Settings Provided by NNMi" on page 354 for information about which Monitored Attributes are available for Interface Groups.
A High Threshold situation occurs when: The *Monitored Attribute* is greater than the *High Value* for at least the time specified in *High Duration* within the *High Duration Window*. When these criteria are met, NNMi does the following: • Updates the Threshold's state value to **High** for the appropriate Interface. • Generates the related incident (if one is Enabled ☑). If an incident is generated, NNMi closes that incident when the Threshold criteria are no longer met.	
High Value	Designate the percentage between 0.00 and 100.00 above which becomes a threshold situation. For special situations, the following values can be used: • 0.000000000000001 (or 1E-15 in Scientific Notation) for the smallest value greater than zero. • 99.99999999999999 for the highest value less than one hundred. The High Value must be greater than or equal to the designated Low Value. **Note:** If you use the highest possible value, the High threshold is disabled because it cannot be *crossed*.
High Value Rearm	The High Value Rearm designates the lower boundary of the High Threshold *range of values*. After entering a High threshold situation, when a returned value is below the specified High Value Rearm, the following happens (for Time-Based Thresholds):

Basic Time-Based Threshold Settings, continued

Attribute	Description
	• The current polling interval does not contribute toward High Duration. • The criteria for High Duration and High Duration Window determine when the High Threshold situation ends. **Note:** The High Value Rearm must be less than or equal to the High Value and greater than or equal to the Low Value Rearm.
High Duration	Designate the minimum time within which the value must remain in the High range before the threshold state changes to High and (optionally) an incident is generated. The High Duration should be equal to or greater than which ever currently configured *Fault Polling Interval* or *Performance Polling Interval* setting is influencing the Monitored Attribute you chose, because that is how often NNMi provides a data point. See the tables in "About Threshold Settings Provided by NNMi" on page 354 for details. See the following topics for instructions about finding the current polling interval setting: • "Default Settings for Monitoring" on page 345 • "Interface Group Settings for Monitoring (NNM iSPI Performance for Metrics)" on page 374 • "Node Group Settings for Monitoring" on page 391 **Tip:** Setting both the High Duration and High Duration Window to zero disables the High threshold.
High Duration Window	Designate the window of time within which the High Duration criteria must be met. The value must be greater than 0 (zero) and can be the same as or greater than the High Duration value. NNMi uses a sliding window, meaning that each time the High Window Duration is reached, NNMi drops the oldest polling interval and adds the most recent. **Tip:** Setting both the High Duration and High Duration Window to zero disables the High threshold.
A Low Threshold situation occurs when: The *Monitored Attribute* is lower than the *Low Value* for at least the time specified in *Low Duration* within the *Low Duration Window*. When these criteria are met, NNMi does the following: • Updates the Threshold's state value to **Low** for the appropriate Interface.	

Basic Time-Based Threshold Settings, continued

Attribute	Description
	• Generates the related incident (if one is Enabled ☑). If an incident is generated, NNMi closes that incident when the Threshold criteria are no longer met.
Low Value	Designate the percentage between 0.00 and 100.00 below which becomes a Low threshold situation. For special situations, the following values can be used: • 0.000000000000001 (or 1E-15 in Scientific Notation) for the smallest value greater than zero. • 99.99999999999999 for the highest value less than one hundred. The Low Value must be less than or equal to the designated High Value. **Note:** If you use zero (the minimum possible value), the Low threshold is disabled because it cannot be *crossed*.
Low Value Rearm	The Low Value Rearm designates the upper boundary of the Low Threshold *range of values*. After entering a Low threshold situation, when a returned value is above the specified Low Value Rearm, the following happens (for Time-Based Thresholds): • The current polling interval does not contribute toward Low Duration. • The criteria for Low Duration and Low Duration Window determine when Low Threshold ends. **Note:** The Low Value Rearm must be greater than or equal to the Low Value and less than or equal to the High Value Rearm.
Low Duration	Designate the minimum time within which the value must remain in the Low range before the threshold state changes to Low and (optionally) an incident is generated. The Low Duration should be equal to or greater than which ever currently configured *Fault Polling Interval* or *Performance Polling Interval* setting is influencing the Monitored Attribute you chose, because that is how often NNMi provides a data point. See the tables in "About Threshold Settings Provided by NNMi" on page 354 for details. See the following topics for instructions about finding the current polling interval setting: • "Default Settings for Monitoring" on page 345 • "Interface Group Settings for Monitoring (NNM iSPI Performance for Metrics)" on page 374 • "Node Group Settings for Monitoring" on page 391

Basic Time-Based Threshold Settings, continued

Attribute	Description
	Tip: Setting both the Low Duration and Low Duration Window to zero disables the Low threshold.
Low Duration Window	Designate the window of time within which the Low Duration criteria must be met. The value must be greater than 0 (zero) and can be the same as or greater than the Low Duration value. NNMi uses a sliding window, meaning that each time the Low Window Duration is reached, NNMi drops the oldest polling interval and adds the most recent. **Tip:** Setting both the Low Duration and Low Duration Window to zero disables the Low threshold.

Configure Baseline Settings for Interface Groups

Use the **Baseline Settings** form to configure NNMi and the NNM iSPI Performance for Metrics for baseline monitoring in your network environment. (See "Purchase an HP Network Node Manager i Smart Plug-in" on page 1486 for more information about the HP Network Node Manager iSPI Performance for Metrics Software.) If you set baseline ranges, you can configure NNMi to generate an Incident when any value is outside of the baseline range.

NNM iSPI Performance for Metrics uses Triple Exponential Smoothing technique to predict the baseline values of a monitored attribute. See "Integrating with Other iSPIs" in the NNM iSPI Performance for MetricsOnline Help for more information about how baseline data is collected. for more information about how baseline data is collected.

NNM iSPI Performance for Metrics provides exception reports to track the frequency of threshold breaches. You can open these reports with **Actions** → **HP NNM iSPI Performance** → **Reporting - Report Menu** in the incident, node, or interface views and forms. (See NNM iSPI Performance for Metrics Actions.)

To establish baseline settings for an Interface Group:

1. Navigate to the **Interface Group Settings** form.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Expand the **Monitoring** folder.
   c. Select **Monitoring Configuration**.
   d. Locate the **Interface Group Settings** tab.
   e. Do one of the following:
      - To create an Interface Group Settings definition, click the ✳ New icon.
      - To edit an Interface Group Settings definition, select a row and click the 📂 Open icon.

- To delete an Interface Group Settings definition, select a row and click the Delete button.

2. Navigate to the **Baseline Settings** tab.
3. Do one of the following:
   - To create an Baseline Settings definition, click the New icon.
   - To edit an Baseline Settings definition, select a row and click the Open icon.
4. Establish the baseline settings (see the Baseline Settings table).
5. Navigate to the **Baseline Deviations Settings** tab.
6. Establish the baseline range for monitoring this Interface Group (see the Baseline Deviations Settings table).
7. By default, NNMi monitors only interfaces that are connected to other interfaces. When SNMP polling is enabled, NNMi automatically detects most connections. See ""Add or Delete a Layer 2 Connection" on page 284" for information about manual overrides.

   Optional. If you want to expand monitoring behavior for this group to include unconnected Interfaces, indicate your choices in the Extend the Scope of Polling Beyond Connected Interfaces group box.
8. Click **Save and Close** to return to the Interface Group Settings form.
9. Click **Save and Close** to return to the Monitoring Configuration form.
10. Click **Save and Close**. NNMi applies your changes. The next regularly scheduled monitoring cycle uses the new settings.

    > **Caution:** When you establish monitoring configuration settings, NNMi must recalculate the settings for all affected objects. This can take some time and slow down your system. Consider making this change during a slow time in your network environment
11. *Optional*. Customize the node monitoring behavior. See "Node Group Settings for Monitoring" on the next page. Also see "Detect Interface Changes" on page 280.
12. See also "Find Threshold Results" on page 414.

Baseline Settings for this Interface Group Setting

Attribute	Description
Monitored Attribute	NNMi gathers data to calculate thresholds. See "About Threshold Settings Provided by NNMi" on page 354 for information about which attributes apply here. **Tip:** You may see attributes in the selection list that do not apply here.
Threshold Enabled	Use this attribute to temporarily disable the threshold: **Enable** □ = Temporarily disable the selected configuration. **Enable** ☑ = Enable the selected configuration.

Baseline Settings for this Interface Group Setting, continued

Attribute	Description
Duration	Designate the minimum time within which the value must remain out of the configured Baseline Range before the state changes to Abnormal Range and (optionally) an incident is generated. Use the **Baseline Deviation Settings** tab to set the upper and lower limits of the baseline range. Note the following: • If you do not configure a Baseline Range, NNMi uses the default value of 3 standard deviations. • The Polling Interval should be less than or equal to the Duration.
Duration Window	Designate the window of time in which the Upper Baseline Limt or Lower Baseline Limit criteria must be met. **Note:** The value must be greater than 0 (zero) and can be the same as the Duration value. NNMi uses a sliding window, meaning that each time the Duration is reached, NNMi drops the oldest polling interval and adds the most recent. See "Examples of Time-Based Threshold Monitoring" on page 368 for more information.

Baseline Deviations Settings for this Interface Group Setting

Attribute	Description
Upper Baseline Limit Enabled	If ☑ enabled, NNMi uses the Upper Baseline Limit attribute value to determine the number of standard deviations above the average that defines the upper baseline limit. If □ disabled: NNMi does not define the upper baseline limit.
Upper Baseline Limit - Deviations above average	Enter the number of standard deviations above the average values that NNMi should use to determine the upper baseline limit.
Lower Baseline Limit Enabled	If ☑ enabled, NNMi uses the Lower Baseline Limit attribute value to determine the number of standard deviations below the average that defines the lower baseline limit. If □ disabled: NNMi does not define the lower baseline limit.
Lower Baseline Limit - Deviations below average	Enter the number of standard deviations below the average values that NNMi should use to determine the lower baseline limit.

Node Group Settings for Monitoring

Before you start, you must establish one or more Node Group definitions that identify the nodes to which these monitoring settings will apply. See also, "Configure Node Group Status" on page 318 and "Node Groups Provided by NNMi" on page 335.

> **Tip:** NNMi administrators can check network latency for a Node Group by adjusting the following for the management addresses associated with the specified group of nodes:
>
> - ICMP polling frequency
> - ICMP echo request packet data payload size
>
> See "Maintaining NNMi" in the HP Network Node Manager i Software Deployment Reference for more information.

To establish monitoring behavior for a predefined Node Group:

1. Navigate to the **Node Group Settings** form.
   1. From the workspace navigation panel, select the **Configuration** workspace.
   2. Expand the **Monitoring** folder.
   3. Select **Monitoring Configuration**.
   4. Locate the **Node Group Settings** tab.
   5. Do one of the following:
      - To create an Node Group Settings definition, click the ✳ New icon.
      - To edit an Node Group Settings definition, select a row and click the 📂 Open icon.
      - To delete an Node Group Settings definition, select a row and click the ✖ Delete button
2. Establish the appropriate settings to identify this Node Setting definition (see Basics table).
3. *Optional*. Configure the Fault Monitoring behavior for this Node Setting definition (see Fault Monitoring table).
4. (*NNM iSPI Performance for Metrics*) If the HP Network Node Manager iSPI Performance for Metrics Software is installed:
   - Configure the Performance Monitoring behavior for this Node Setting definition. See Performance Monitoring table.
   - Configure the Baseline Settings. Navigate to the Baseline Settings tab. See "Configure Baseline Settings for Node Groups" on page 410.
5. *Optional*. Set thresholds.Navigate to the Threshold Settings tab. See "Configure Threshold Monitoring for Node Groups" on page 402 for more information.

   When you configure thresholds using this technique, NNMi uses the assigned Node Group as a filter (only monitoring the threshold for devices that belong to the specified Node Group). The thresholds you configure here can be monitoring node components (such as disk space utilization) or monitoring interfaces within nodes (such as input error rate).
6. By default, NNMi monitors only interfaces that are connected to other interfaces. When SNMP polling is enabled, NNMi automatically detects most connections. See "Add or Delete a Layer 2 Connection" on page 284 for information about manual overrides.

Optional. If you want to expand monitoring behavior for this group to include unconnected Interfaces, indicate your choices in the Extend the Scope of Polling Beyond Connected Interfaces group box.

7. *Optional*. Configure the Default Change Detection Monitoring (see Default Change Detection Monitoring table).

8. Click **Save and Close** to return to the Monitoring Configuration form.

9. Click **Save and Close**. NNMi applies your changes. The next regularly scheduled monitoring cycle uses the new settings.

> **Caution:** When you establish monitoring configuration settings, NNMi must recalculate the settings for all affected objects. This can take some time and slow down your system. Consider making this change during a slow time in your network environment.

To verify that State Poller is working as expected, see **Help** → **System Information** and select the the **State Poller** tab. NNMi displays a report with current details about the State Poller process.

Optional. Customize the interface monitoring behavior. See "Interface Group Settings for Monitoring (NNM iSPI Performance for Metrics)" on page 374 .

Basics

Attribute	Description
Ordering	Enter a unique string (any length), characters 0 through 9. Consider using increments of 100 for the flexibility to insert additional items between existing items over time. NNMi decides which monitoring configurations apply to a node or interface based on the ordering number assigned to the configuration definitions. NNMi monitors the device according to the first match (checked from lowest number to highest number within each category). Categories are read in sequence. Click here for a description of the sequence. 1. **Interface Group Settings**: NNMi monitors each of the Node's Interfaces and IP Addresses based on the first matching Interface Group Settings definition. The first match is the Interface Group Settings definition with the lowest Ordering number, then Baseline Settings. 2. **Node Group Settings**: NNMi monitors each Node and each previously unmatched Interface or IP Address based on the first matching Node Group Settings definition. The first match is the Node Group Settings definition with the lowest Ordering number, then Baseline Settings. **Note:** Child node groups are included in the Ordering hierarchy. This means that if the parent node group has a lower Ordering number (for example, parent=10, child=20), then the monitoring configuration specified for the parent node group also applies to the nodes in the child node group. To override a parent node group monitoring configuration, set the Ordering number for the child node group to a number that is

Basics, continued

Attribute	Description
	lower than the parent (for example, parent=20, child=10). 3. **Default Settings**: If no match is found for a Node, Interface, or IP Address in 1 or 2, NNMi applies the default Monitoring Configuration settings. No duplicate Ordering numbers are permitted. Each Node Setting ordering number must be unique.
Node Group	Choose one predefined Node Group from the list. See "Create Node Groups" on page 295 for more information. (*NNMi Advanced with IPv6 enabled*) See also "Node Groups of IPv4 or IPv6 Addresses " on page 306.
Enable SNMP Polling on Node	If ☑ enabled, NNMi contacts the SNMP Agent on each node in the specified Node Group to gather SNMP data for monitoring purposes. If □ disabled, NNMi does not contact the SNMP Agent on nodes in the specified Node Group for monitoring purposes (does not generate SNMP traffic to the nodes). **Note:** If you use Auto-Discovery, NNMi might detect Nodes and add them to the NNMi database as non-SNMP nodes. To configure Auto-Discovery to not add specified IP addresses to the NNMi database, not acknowledge any Hints received about them, nor gather Discovery Hints from them unless the address is a discovery seed, see "Set Outside Limits for Auto-Discovery" on page 228.

Fault Monitoring

Attribute	Description
ICMP Fault Monitoring: Enable Management Address Polling	If ☑ enabled, State Poller only issues ICMP (ping) requests to the management address for a node. **Note:** In the Global Control section of the Monitoring Configuration form, the Enable State Polling attribute must be enabled, too. If □ disabled, State Poller does one of the following: • If neither this attribute nor *Enable ICMP Fault Polling* is selected, State Poller does not use ICMP to monitor nodes covered by this configuration setting. • If *Enable ICMP Fault Polling* is selected, State Poller uses ICMP to monitor ALL IP addresses covered by this configuration setting.
ICMP Fault	If ☑ enabled, State Poller issues ICMP (ping) requests to verify the

Fault Monitoring, continued

Attribute	Description
Monitoring: Enable IP Address Fault Polling **Note:** This monitoring option is useful for devices that do not support SNMP. By default, this feature is enabled for the "Non-SNMP Devices" Node Group.	availability of discovered IP address. **Note:** In the Global Control section of this form, the Enable State Polling attribute must be enabled, too. If □ disabled, State Poller does the following: • If neither this attribute nor *Management IP Address Polling* is selected, State Poller does not use ICMP to monitor nodes covered by this configuration setting. • IP addresses (both previously discovered and newly discovered) have a State attribute value of "Not Polled" and a Status attribute value of "No Status" with the color of the IP address map-symbol set to beige. See Layer 3 Neighbor View. • If both ICMP and SNMP are disabled for a Node, the Node has a Status attribute value of "No Status" and the color of the Node map-symbol background shape is set to beige. **Tip:** To turn off ICMP polling within a subset of your network environment, use the Communication Configuration workspace Region definitions. You can define your own Regions that identify any unreachable addresses in your management domain (for example, the **private IP addresses**[1]).
SNMP Fault Monitoring: Enable Interface Fault Polling	If ☑ enabled, State Poller monitors all interfaces by issuing SNMP read-only queries to devices assigned to this level of the monitoring hierarchy. By default, any connected interface is monitored for MIB-II ifAdminStatus and ifOperStatus. (ifAdminStatus is set by the device administrator. ifOperStatus indicates the operational status of interface health.) If you have *unconnected* interfaces that you want to monitor, expand NNMi monitoring behavior with the Poll Unconnected Interfaces and the Poll Interfaces Hosting IP Addresses attributes. **Note:** The following attributes must also be enabled: • In the Global Control section of this form, the Enable State Polling attribute must be enabled, too. See Layer 2 Neighbor View. (See "Global Control Settings for Monitoring" on page 343 for more information.) • In the Communication Configuration view, enable State Poller queries

[1] These are IPv4 addresses that can be reused in home and office local area networks (LANs). Following the standards set by RFC 1918 and RFC 4193 (10.*.*.*, 169.254.*.*, 172.16-31.*.*, and 192.168.*.*)

Fault Monitoring, continued

Attribute	Description
	with the applicable Enable SNMP Communication attributes (see "Configuring Communication Protocol" on page 119 for more information). If □ disabled, for devices assigned to this level of the monitoring hierarchy: • Causal Engine calculates Status based only on IP address State. • The Interface objects previously discovered change to a State attribute value of "Not Polled" and a Status attribute value of "No Status" (plus any related map-symbol changes to a beige color).
SNMP Fault Monitoring: Enable Card Fault Polling	Use this attribute to poll fault metrics for cards. Card fault metrics include Administrative State, Operational State, and Standby State. **Note:** Card Fault Polling is enabled by default. If ☑ enabled, NNMi gathers fault data related to the card fault metrics in devices assigned to this level of the monitoring hierarchy. If □ disabled, NNMi does not extend data collection behavior to include card fault data about devices assigned to this level of the monitoring hierarchy. **Tip:** NNMi uses the same polling interval set for the Fault Polling Interval.
SNMP Fault Monitoring: Enable Node Component Fault Polling **Note:** By default, this feature is enabled for the "Routers" and "Networking Infrastructure Devices" Node Groups.	Use this attribute to poll Node Component fault metrics. Node Component fault metrics include the following: Fan, Power Supply, Temperature, and Voltage. **Note:** Node Component Fault Polling is disabled by default. Only the health of the Power Supply and Fan Node Components are propagated to the Node level. If ☑ enabled, NNMi gathers fault data related to the Node Component fault metrics in devices assigned to this level of the monitoring hierarchy. If □ disabled, NNMi does not extend data collection behavior to include Node Component fault data about devices assigned to this level of the monitoring hierarchy. **Tip:** NNMi uses the same polling interval set for the Fault Polling Interval.
Fault Polling Interval	The time that State Poller waits between issuing queries to gather information for any of the following that are enabled: ICMP Polling,

Fault Monitoring, continued

Attribute	Description
	SNMP Polling, Poll Unconnected Interfaces, and Poll Interfaces Hosting IP addresses. The default Fault Polling Interval is 5 minutes, except for the Node Group named Microsoft Windows Systems which is 10 minutes. **Note:** NNMi monitors SNMP agents (Management Addresses) according to this Fault Polling Interval, *even if ICMP Polling, SNMP Polling, Poll Unconnected Interfaces, and Poll Interfaces Hosting IP addresses are all disabled*. To prevent an SNMP Agent's address from being monitored, one of the following must be true: State Polling is disabled, current Communication Configuration settings turn off SNMP for the SNMP agent's address, the parent Node is set to Not Managed or Out of Service, or the parent node belongs to a Monitoring Configuration's Node Group with □ **Enable SNMP Polling on Node** disabled.

SNMP Performance Monitoring (*NNM iSPI Performance for Metrics*)

Attribute	Description
LAN Performance Monitoring: Enable Interface Performance Polling	(*NNM iSPI Performance for Metrics*) Use this attribute to extend the range of polling data that NNMi collects. HP Network Node Manager iSPI Performance for Metrics Software uses the additional data in a series of performance reports. See "Purchase an HP Network Node Manager i Smart Plug-in" on page 1486 for more information. When enabled, network traffic increases on your network because NNMi gathers performance data about each member of this group on a regular schedule. If ☑ enabled, NNMi gathers basic Interface performance data from Interfaces in devices assigned to this level of the monitoring hierarchy. If □ disabled, NNMi does not extend data collection behavior to include performance data about Interfaces assigned to this level of the monitoring hierarchy. **Note:** The Enable State Polling field must be enabled, too. By default the performance of connected interfaces and addresses is monitored. If you have *unconnected* interfaces that you want to monitor, expand NNMi monitoring behavior by enabling Poll Unconnected Interfaces.
WAN Performance Monitoring: Enable DSx Interface Performance Polling	(*NNM iSPI Performance for Metrics*) Use this attribute to extend the range of polling data that NNMi collects. HP Network Node Manager iSPI Performance for Metrics Software uses the additional data in a series of performance reports. When enabled, network traffic increases on your network because NNMi gathers performance data about each member of

SNMP Performance Monitoring (NNM iSPI Performance for Metrics), continued

Attribute	Description
	the **DSx Interfaces** interface group on a regular schedule. See "Interface Groups Provided by NNMi" on page 338 for more information. If ☑ enabled, NNMi gathers DSx performance data from DSx Interfaces assigned to this level of the monitoring hierarchy. If □ disabled, NNMi does not gather DSx performance data from DSx Interfaces assigned to this level of the monitoring hierarchy.
WAN Performance Monitoring: Enable SONET Interface Performance Polling	(*NNM iSPI Performance for Metrics*) Use this attribute to extend the range of polling data that NNMi collects. HP Network Node Manager iSPI Performance for Metrics Software uses the additional data in a series of performance reports. When enabled, network traffic increases on your network because NNMi gathers performance data about each member of the **SONET Interfaces** interface group on a regular schedule. See "Interface Groups Provided by NNMi" on page 338 for more information. If ☑ enabled, NNMi gathers SONET performance data from SONET Interfaces assigned to this level of the monitoring hierarchy. If □ disabled, NNMi does not gather SONET performance data from SONET Interfaces assigned to this level of the monitoring hierarchy.
WAN Performance Monitoring: Enable ATM Interface Performance Polling	(*NNM iSPI Performance for Metrics*) Use this attribute to extend the range of polling data that NNMi collects. HP Network Node Manager iSPI Performance for Metrics Software uses the additional data in a series of performance reports. When enabled, network traffic increases on your network because NNMi gathers performance data for each ATM Interface. If ☑ enabled, NNMi gathers ATM performance data from ATM Interfaces assigned to this level of the monitoring hierarchy. If □ disabled, NNMi does not gather ATM performance data from ATM Interfaces assigned to this level of the monitoring hierarchy. **Note:** • This option gathers metrics from ATM-MIB and CISCO-AAL5-MIB. • See also "Configure Discovery of ATM/Frame Relay Interfaces" on page 201.
WAN Performance Monitoring: Enable Frame Relay Interface Performance Polling	(*NNM iSPI Performance for Metrics*) Use this attribute to extend the range of polling data that NNMi collects. HP Network Node Manager iSPI Performance for Metrics Software uses the additional data in a series of performance reports. When enabled, network traffic increases on your network because NNMi gathers performance data for each Frame Relay Interface.

SNMP Performance Monitoring (NNM iSPI Performance for Metrics), continued

Attribute	Description
	If ☑ enabled, NNMi gathers Frame Relay performance data from Frame Relay Interfaces assigned to this level of the monitoring hierarchy. If □ disabled, NNMi does not gather Frame Relay performance data from Frame Relay Interfaces assigned to this level of the monitoring hierarchy. This option gathers the following types of metrics: • Circuit in and out octets, errors, and discards • Committed Information Rate (CIR) and Extended Information Rate (EIR) utilization • Forward Error Congestion Notification (FECN) and Backward Error Congestion Notification (BECN) counts See also "Configure Discovery of ATM/Frame Relay Interfaces" on page 201.
Component Performance Monitoring: Enable Node Component Performance Polling **Note:** By default, this feature is enabled for the "Routers" Node Group if HP Network Node Manager iSPI Performance for Metrics Software is installed.	(*NNM iSPI Performance for Metrics*) Use this attribute to poll Node Component performance. An NNMi administrator can set the threshold for node components related to the following performance metrics: CPU utilization, memory utilization, buffer utilization, buffer miss rate, and buffer failure rate. **Note:** Node Component Performance Polling is disabled by default. If ☑ enabled, NNMi gathers performance data related to the Node Component performance metrics in devices assigned to this level of the monitoring hierarchy. If □ disabled, NNMi does not extend data collection behavior to include Node Component performance data about devices assigned to this level of the monitoring hierarchy. **Tip:** NNMi uses the same polling interval set for the Performance Polling Interval.
Performance Polling Interval	(*NNM iSPI Performance for Metrics*) Use this field to set the time period that NNMi waits between issuing network traffic to gather performance data for the HP Network Node Manager iSPI Performance for Metrics Software. The default Performance Polling Interval is 5 minutes, except for the Node Group named Microsoft Windows Systems which is 10 minutes.

Extend the Scope of Polling Beyond Connected Interfaces

Attribute	Description
Poll Unconnected Interfaces	If ☑ enabled, NNMi monitors all interfaces within discovered devices (both connected and unconnected). All interfaces are monitored for MIB-II ifAdminStatus and ifOperStatus. (ifAdminStatus is set by the device administrator. ifOperStatus indicates the operational status of interface health.) **Note:** The Enable State Polling field must be enabled, and SNMP polling of some type must be enabled (for example, Enable SNMP Fault Monitoring and Enable SNMP Performance Polling). If □ disabled, State Poller polls according to other configuration settings. **Tip:** Your discovery configuration choices might need to be adjusted to get the results you want. For example, to meet the “connected” criteria for interfaces in switches that do not have an IP address you must add the device to which the interface is connected as a discovery seed. See"Specify Discovery Seeds" on page 256.
Poll Interfaces Hosting IP Addresses **Note:** This monitoring option is useful for Router interfaces. By default, this feature is enabled for the "Routers" Node Group.	If ☑ enabled, any unconnected interface that has one or more addresses associated with it is monitored for MIB-II ifAdminStatus and ifOperStatus. (ifAdminStatus is set by the device administrator. ifOperStatus indicates the operational status of interface health.) **Note:** The Enable State Polling field must be enabled, and SNMP polling of some type must be enabled (for example, Enable SNMP Fault Monitoring and Enable SNMP Performance Polling). By monitoring the Interface (in addition to the IP address), NNMi can make more informed decisions about the health of each IP address associated with an unconnected interface. If □ disabled, State Poller polls according to other configuration settings. **Tip:** The Communication Configuration workspace provides a method of overriding this setting for specific Regions. You can

Extend the Scope of Polling Beyond Connected Interfaces , continued

Attribute	Description
Poll Unconnected Interfaces	If ☑ enabled, NNMi monitors all interfaces within discovered devices (both connected and unconnected). All interfaces are monitored for MIB-II ifAdminStatus and ifOperStatus. (ifAdminStatus is set by the device administrator. ifOperStatus indicates the operational status of interface health.) **Note:** The Enable State Polling field must be enabled, and SNMP polling of some type must be enabled (for example, Enable SNMP Fault Monitoring and Enable SNMP Performance Polling). If □ disabled, State Poller polls according to other configuration settings. **Tip:** Your discovery configuration choices might need to be adjusted to get the results you want. For example, to meet the “connected” criteria for interfaces in switches that do not have an IP address you must add the device to which the interface is connected as a discovery seed. See"Specify Discovery Seeds" on page 256.
	define your own Region to easily turn off polling to any unreachable addresses in your management domain (for example, the **private IP addresses**[1]).

Default Change Detection Monitoring

Attribute	Description
Enable Number of Interfaces (ifNumber) Polling	**Tip:** For more information, see "Detect Interface Changes" on page 280. When enabled ☑, NNMi polls for the number of interfaces using the `ifNumber` value for the node. If the number of interfaces has changed, NNMi initiates a rediscovery of the node. Polling is suspended until the discovery is complete. When disabled □, NNMi does not actively poll for a change in the number of interfaces. The change is detected the next time the node is rediscovered.

[1] These are IPv4 addresses that can be reused in home and office local area networks (LANs). Following the standards set by RFC 1918 and RFC 4193 (10.*.*.*, 169.254.*.*, 172.16-31.*.*, and 192.168.*.*)

Default Change Detection Monitoring, continued

Attribute	Description
Enable Entity (entLastChangeTime) Polling	When enabled ☑, NNMi polls for the last change time from the ENTITY-MIB `entLastChangeTime` value. If the time has changed, NNMi initiates a rediscovery of the node. Polling is suspended until the discovery is complete. When disabled □, NNMi does not actively poll the `entLastChangeTime` MIB value. The change is detected the next time the node is rediscovered.
Change Detection Polling Interval	The time that State Poller waits between issuing queries to gather information for the Number of Interfaces (`ifNumber`) and Entity Change (`entLastChangeTime`) settings enabled for Change Detection Monitoring. The default Change Detection Polling Interval is 4 hours.

Configure Threshold Monitoring for Node Groups

If you set thresholds, NNMi generates an Incident when any threshold is violated.

You can set node thresholds using either of the following methods:

- "Configure Count-Based Threshold Monitoring for Node Groups" below
- "Configure Time-Based Threshold Monitoring for Node Groups" on page 406

Related Topics

"About Threshold Settings Provided by NNMi" on page 354

"Threshold Monitoring Behavior After a System Restart or Configuration Change" on page 415

Configure Count-Based Threshold Monitoring for Node Groups

Count-Based Threshold Settings enable you to determine as soon as a threshold is reached (for example, the CPU utilization for a node reaches 90%).

To establish count-based threshold monitoring behavior for nodes:

1. *Prerequisite*. Before setting thresholds, analyze performance data over time to determine wise threshold settings for each Node Group. For more information, see the following topics:
   - "Determine Reasonable Threshold Settings" on page 413.
   - "Examples of Count-Based Threshold Monitoring" on page 364 .
2. Navigate to the **Node Group Settings** form.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Expand the **Monitoring** folder.
   c. Select **Monitoring Configuration**.

   d. Navigate to the **Node Group Settings** tab.

   e. Do one of the following:

      ○ To create an Node Group Settings definition, click the New icon.

      ○ To edit an Node Group Settings definition, select a row and click the Open icon.

3. In the **Node Group Settings** form, navigate to the **Threshold Settings** tab.

4. Do one of the following:

   - To create a threshold definition, click the New icon and select **Count-Based Threshold Settings**.

   - To edit a threshold definition, select a row and click the Open icon.

   - To delete a threshold definition, select a row and click the Delete icon.

5. Select the Monitored Attribute you want to monitor and establish the threshold values for that attribute (see Basic Count-Based Threshold Settings table).

   When you configure thresholds using this technique, NNMi uses the assigned Node Group as a filter (only monitoring the threshold for nodes belonging to the specified Node Group).

6. Click **Save and Close** to return to the **Node Group Settings** form.

7. Click **Save and Close** to return to the **Monitoring Configuration** form.

8. Click **Save and Close**. NNMi applies your changes during the next regularly scheduled monitoring cycle.

   > **Note:** Threshold Incidents are disabled by default within NNMi to prevent Incident storms. If you are ready to generate Threshold Incidents, see "Generate Performance Threshold Incidents (NNM iSPI Performance for Metrics)" on page 764. See also "Custom Incident Attributes Provided by NNMi (Information for Administrators)" on page 647 for a description of the special custom incident attributes available in Threshold Incidents.

9. See also "Find Threshold Results" on page 414.

Basic Count-Based Threshold Settings

Attribute	Description
Monitored Attribute	In the Monitored Attribute drop-down list, select the attribute for which you want to establish a threshold configuration. **Tip:** Some of the choices in the Monitored Attribute selection list do not apply in this context. See the tables in "About Threshold Settings Provided by NNMi" on page 354 for information about which Monitored Attributes are available for Node Groups.
A High Threshold situation occurs when:	

Basic Count-Based Threshold Settings, continued

Attribute	Description
The *Monitored Attribute* is greater than the *High Value* for *High Trigger Count* cycles. When these criteria are met, NNMi does the following: • Updates the Threshold's state value to **High** for the appropriate Interface. • Generates the related incident (if one is Enabled ☑). If an incident is generated, NNMi closes that incident when the Threshold criteria are no longer met.	
High Value	Designate the value that above which becomes a threshold situation. Use one of the following: • Designate a percentage between 0.00 and 100.00. For special situations, the following values can be used: ▪ 0.000000000000001 (or 1E-15 in Scientific Notation) for the smallest value greater than zero. ▪ 99.99999999999999 for the highest value less than one hundred. • Designate any appropriate integer value (for example, a Management Address ICMP Response Time of 0 or greater milliseconds). The High Value must be greater than or equal to the designated Low Value. **Note:** If you use the highest possible value, the threshold is disabled because it cannot be *crossed*.
High Value Rearm	The High Value Rearm designates the lower boundary of the High Threshold *range of values*. After entering a High threshold situation, when a returned value is below the specified High Value Rearm, the High Threshold situation ends (for Count-Based Thresholds). **Note:** The High Value Rearm must be less than or equal to the High Value and greater than or equal to the Low Value Rearm.
High Trigger Count	Designate the number of consecutive polling intervals the returned value must be greater than the specified High Value to meet the High Threshold criteria. The default value is 1. **Tip:** If the polled value represents an average over the configured polling interval, a trigger count of 1 is often appropriate. See the currently configured *Fault Polling Interval* or *Performance Polling Interval* setting that is influencing the Monitored Attribute you chose, because that is how often NNMi provides a data point. See the tables in "About Threshold Settings Provided by NNMi" on page 354 for details. See the

Basic Count-Based Threshold Settings, continued

Attribute	Description
	following topics for instructions about finding the current polling interval setting: • "Default Settings for Monitoring" on page 345 • "Interface Group Settings for Monitoring (NNM iSPI Performance for Metrics) " on page 374 • "Node Group Settings for Monitoring" on page 391
A Low Threshold situation occurs when: The *Monitored Attribute* is less than the *Low Value* for *Low Trigger Count* cycles. When these criteria are met, NNMi does the following: • Updates the Threshold's state value to **Low** for the appropriate Interface. • Generates the related incident (if one is Enabled ☑). If an incident is generated, NNMi closes that incident when the Threshold criteria are no longer met.	
Low Value	Designate the value that below which becomes a threshold situation. Use one of the following: • Designate a percentage between 0.00 and 100.00. For special situations, the following values can be used: ▪ 0.000000000000001 (or 1E-15 in Scientific Notation) for the smallest value greater than zero. ▪ 99.99999999999999 for the highest value less than one hundred. • Designate any appropriate integer value (for example, a Management Address ICMP Response Time of 0 or greater milliseconds). The Low Value must be less than or equal to the designated High Value. **Note:** If you use the minimum possible value, the Low threshold is disabled because it cannot be *crossed*.
Low Value Rearm	The Low Value Rearm designates the upper boundary of the Low Threshold *range of values*. After entering a Low threshold situation, when a returned value is above the specified Low Value Rearm, the Low Threshold situation ends (for Count-Based Thresholds). **Note:** The Low Value Rearm must be greater than or equal to the Low Value and less than or equal to the High Value Rearm.
Low Trigger Count	Designate the number of consecutive polling interval the returned value must be less than the specified Low Value to meet the Low Threshold criteria. The

Basic Count-Based Threshold Settings, continued

Attribute	Description
	default value is 1. **Tip:** If the polled value represents an average over the configured polling interval, a trigger count of 1 is often appropriate. See the currently configured *Fault Polling Interval* or *Performance Polling Interval* setting that is influencing the Monitored Attribute you chose, because that is how often NNMi provides a data point. See the tables in "About Threshold Settings Provided by NNMi" on page 354 for details. See the following topics for instructions about finding the current polling interval setting: • "Default Settings for Monitoring" on page 345 • "Interface Group Settings for Monitoring (NNM iSPI Performance for Metrics) " on page 374 • "Node Group Settings for Monitoring" on page 391

Configure Time-Based Threshold Monitoring for Node Groups

Time-Based Threshold Settings enable you to determine whether a threshold is reached for a particular duration of time (for example, the CPU utilization for a node is above 90 percent for 20 out of 30 minutes).

To establish time-based threshold monitoring behavior for nodes:

1. *Prerequisite*. Before setting thresholds, analyze performance data over time to determine wise threshold settings for each Node Group.
   - "Determine Reasonable Threshold Settings" on page 413.
   - "Examples of Time-Based Threshold Monitoring" on page 368 .
2. Navigate to the **Node Group Settings** form.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Expand the **Monitoring** folder.
   c. Select **Monitoring Configuration**.
   d. Navigate to the **Node Group Settings** tab.
   e. Do one of the following:
      - To create an Node Group Settings definition, click the ✱ New icon.
      - To edit an Node Group Settings definition, select a row and click the 📂 Open icon.
3. In the **Node Group Settings** form, navigate to the **Threshold Settings** tab.
4. Do one of the following:

- To create a threshold definition, click the New icon and select **Time-Based Threshold Settings**.
- To edit a threshold definition, select a row and click the Open icon.
- To delete a threshold definition, select a row and click the Delete icon.

5. Select the Monitored Attribute you want to monitor and establish the threshold values for that attribute (see Basic Time-Based Threshold Settings table).

   When you configure thresholds using this technique, NNMi uses the assigned Node Group as a filter (only monitoring the threshold for nodes belonging to the specified Node Group).

6. Click **Save and Close** to return to the **Node Group Settings** form.
7. Click **Save and Close** to return to the **Monitoring Configuration** form.
8. Click **Save and Close**. NNMi applies your changes during the next regularly scheduled monitoring cycle.

   > **Note:** Threshold Incidents are disabled by default within NNMi to prevent Incident storms. If you are ready to generate Threshold Incidents, see "Generate Performance Threshold Incidents (NNM iSPI Performance for Metrics)" on page 764. See also "Custom Incident Attributes Provided by NNMi (Information for Administrators)" on page 647 for a description of the special custom incident attributes available in Threshold Incidents.

9. See also "Find Threshold Results" on page 414.

Basic Time-Based Threshold Settings

Attribute	Description
Monitored Attribute	In the Monitored Attribute drop-down list, select the attribute for which you want to establish a threshold configuration. **Tip:** Some of the choices in the Monitored Attribute selection list do not apply in this context. See the tables in "About Threshold Settings Provided by NNMi" on page 354 for information about which Monitored Attributes are available for Node Groups.
A High Threshold situation occurs when: The *Monitored Attribute* is greater than the *High Value* for at least the time specified in *High Duration* within the *High Duration Window*. When these criteria are met, NNMi does the following: • Updates the Threshold's state value to **High** for the appropriate Interface. • Generates the related incident (if one is Enabled ☑). If an incident is generated, NNMi closes that incident when the Threshold criteria are no longer met.	
High Value	Designate the value that above which becomes a threshold situation. Use one of

Basic Time-Based Threshold Settings, continued

Attribute	Description
	the following: • Designate a percentage between 0.00 and 100.00. For special situations, the following values can be used: ▪ 0.000000000000001 (or 1E-15 in Scientific Notation) for the smallest value greater than zero. ▪ 99.99999999999999 for the highest value less than one hundred. • Designate any appropriate integer value (for example, a Management Address ICMP Response Time of 0 or greater milliseconds). The High Value must be greater than or equal to the designated Low Value. **Note:** If you use the highest possible value, the High threshold is disabled because it cannot be *crossed*.
High Value Rearm	The High Value Rearm designates the lower boundary of the High Threshold *range of values*. After entering a High threshold situation, when a returned value is below the specified High Value Rearm, the following happens (for Time-Based Thresholds): • The current polling interval does not contribute toward High Duration. • The criteria for High Duration and High Duration Window determine when the High Threshold situation ends. **Note:** The High Value Rearm must be less than or equal to the High Value and greater than or equal to the Low Value Rearm.
High Duration	Designate the minimum time within which the value must remain in the High range before the threshold state changes to High and (optionally) an incident is generated. The High Duration should be equal to or greater than which ever currently configured *Fault Polling Interval* or *Performance Polling Interval* setting is influencing the Monitored Attribute you chose, because that is how often NNMi provides a data point. See the tables in "About Threshold Settings Provided by NNMi" on page 354 for details. See the following topics for instructions about finding the current polling interval setting: • "Default Settings for Monitoring" on page 345 • "Interface Group Settings for Monitoring (NNM iSPI Performance for Metrics)" on page 374 • "Node Group Settings for Monitoring" on page 391

Basic Time-Based Threshold Settings, continued

Attribute	Description
	Tip: Setting both the High Duration and High Duration Window to zero disables the High threshold.
High Duration Window	Designate the window of time within which the High Duration criteria must be met. The value must be greater than 0 (zero) and can be the same as or greater than the High Duration value. NNMi uses a sliding window, meaning that each time the High Window Duration is reached, NNMi drops the oldest polling interval and adds the most recent. **Tip:** Setting both the High Duration and High Duration Window to zero disables the High threshold.
A Low Threshold situation occurs when: The *Monitored Attribute* is lower than the *Low Value* for at least the time specified in *Low Duration* within the *Low Duration Window*. When these criteria are met, NNMi does the following: • Updates the Threshold's state value to **Low** for the appropriate Interface. • Generates the related incident (if one is Enabled ☑). If an incident is generated, NNMi closes that incident when the Threshold criteria are no longer met.	
Low Value	Designate the value that below which becomes a threshold situation. Use one of the following: • Designate a percentage between 0.00 and 100.00. For special situations, the following values can be used: ▪ 0.000000000000001 (or 1E-15 in Scientific Notation) for the smallest value greater than zero. ▪ 99.99999999999999 for the highest value less than one hundred. • Designate any appropriate integer value (for example, a Management Address ICMP Response Time of 0 or greater milliseconds). The Low Value must be less than or equal to the designated High Value. **Note:** If you use the minimum possible value, the Low threshold is disabled because it cannot be *crossed*.
Low Value Rearm	The Low Value Rearm designates the upper boundary of the Low Threshold *range of values*. After entering a Low threshold situation, when a returned value is above the

Basic Time-Based Threshold Settings, continued

Attribute	Description
	specified Low Value Rearm, the following happens (for Time-Based Thresholds): • The current polling interval does not contribute toward Low Duration. • The criteria for Low Duration and Low Duration Window determine when Low Threshold ends. **Note:** The Low Value Rearm must be greater than or equal to the Low Value and less than or equal to the High Value Rearm.
Low Duration	Designate the minimum time within which the value must remain in the Low range before the threshold state changes to Low and (optionally) an incident is generated. The Low Duration should be equal to or greater than which ever currently configured *Fault Polling Interval* or *Performance Polling Interval* setting is influencing the Monitored Attribute you chose, because that is how often NNMi provides a data point. See the tables in "About Threshold Settings Provided by NNMi" on page 354 for details. See the following topics for instructions about finding the current polling interval setting: • "Default Settings for Monitoring" on page 345 • "Interface Group Settings for Monitoring (NNM iSPI Performance for Metrics)" on page 374 • "Node Group Settings for Monitoring" on page 391 **Tip:** Setting both the Low Duration and Low Duration Window to zero disables the Low threshold.
Low Duration Window	Designate the window of time within which the Low Duration criteria must be met. The value must be greater than 0 (zero) and can be the same as or greater than the Low Duration value. NNMi uses a sliding window, meaning that each time the Low Window Duration is reached, NNMi drops the oldest polling interval and adds the most recent. **Tip:** Setting both the Low Duration and Low Duration Window to zero disables the Low threshold.

Configure Baseline Settings for Node Groups

Use the **Baseline Settings** form to configure NNMi and the NNM iSPI Performance for Metrics for baseline monitoring in your network environment. (See "Purchase an HP Network Node Manager i Smart Plug-in" on page 1486 for more information about the HP Network Node Manager iSPI Performance for Metrics Software.) If you set baseline ranges, you can configure NNMi to generate an Incident when any value is outside of the baseline range.

NNM iSPI Performance for Metrics uses Triple Exponential Smoothing technique to predict the baseline values of a monitored attribute. See "Integrating with Other iSPIs" in the NNM iSPI Performance for MetricsOnline Help for more information about how baseline data is collected. for more information about how baseline data is collected.

NNM iSPI Performance for Metrics provides exception reports to track the frequency of threshold breaches. You can open these reports with **Actions** → **HP NNM iSPI Performance** → **Reporting - Report Menu** in the incident, node, or interface views and forms. (See NNM iSPI Performance for Metrics Actions.)

To establish baseline settings for the Node Components in a Node Group:

1. Navigate to the **Node Group Settings** form.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Expand the **Monitoring** folder.
   c. Select **Monitoring Configuration**.
   d. Navigate to the **Node Group Settings** tab.
   e. Do one of the following:
      - To create an Node Group Settings definition, click the ✱ New icon.
      - To edit an Node Group Settings definition, select a row and click the Open icon.
2. Navigate to the **Baseline Settings** tab.
3. Do one of the following:
   - To create an Baseline Settings definition, click the ✱ New icon.
   - To edit an Baseline Settings definition, select a row and click the Open icon.
4. Establish the baseline settings (see the Baseline Settings table).
5. Navigate to the **Baseline Deviations Settings** tab.
6. Establish the baseline range for monitoring this Node Group (see the Baseline Deviation Settings table).
7. By default, NNMi monitors only interfaces that are connected to other interfaces. When SNMP polling is enabled, NNMi automatically detects most connections. See "Add or Delete a Layer 2 Connection" on page 284 for information about manual overrides.

   Optional. If you want to expand monitoring behavior for this group to include unconnected Interfaces, indicate your choices in the Extend the Scope of Polling Beyond Connected Interfaces group box.
8. Click **Save and Close** to return to the Node Group Settings form.
9. Click **Save and Close** to return to the Monitoring Configuration form.
10. Click **Save and Close**. NNMi applies your changes. The next regularly scheduled monitoring cycle uses the new settings.

> **Caution:** When you establish monitoring configuration settings, NNMi must recalculate

> the settings for all affected objects. This can take some time and slow down your system. Consider making this change during a slow time in your network environment

11. *Optional*. Customize the node monitoring behavior. See "Node Group Settings for Monitoring" on page 391. Also see "Detect Interface Changes" on page 280.

12. See also "Find Threshold Results" on page 414.

Baseline Settings for this Node Group Setting

Attribute	Description
Monitored Attribute	NNMi gathers data to calculate thresholds. See "About Threshold Settings Provided by NNMi" on page 354 for information about which attributes apply here. **Tip:** You may see attributes in the selection list that do not apply here.
Threshold Enabled	Use this attribute to temporarily disable the threshold: **Enable** ☐ = Temporarily disable the selected configuration. **Enable** ☑ = Enable the selected configuration.
Duration	Designate the minimum time within which the value must remain out of the configured Baseline Range before the state changes to Abnormal Range and (optionally) an incident is generated. Use the **Baseline Deviation Settings** tab to set the upper and lower limits of the baseline range. Note the following: • If you do not configure a Baseline Range, NNMi uses the default value of 3 standard deviations. • The Polling Interval should be less than or equal to the Duration.
Duration Window	Designate the window of time in which the Upper Baseline Limt or Lower Baseline Limit criteria must be met. **Note:** The value must be greater than 0 (zero) and can be the same as the Duration value. NNMi uses a sliding window, meaning that each time the Duration is reached, NNMi drops the oldest polling interval and adds the most recent. See "Examples of Time-Based Threshold Monitoring" on page 368 for more information.

Baseline Deviation Settings for this Node Group Setting

Attribute	Description
Upper Baseline Limit Enabled	If ☑ enabled, NNMi uses the Upper Baseline Limit attribute value to determine the number of standard deviations above the average that defines the upper baseline limit. If ☐ disabled: NNMi does not define the upper baseline limit.

Baseline Deviation Settings for this Node Group Setting, continued

Attribute	Description
Upper Baseline Limit	Enter the number of standard deviations above the average values that NNMi should use to determine the upper baseline limit.
Lower Baseline Limit Enabled	If ☑ enabled, NNMi uses the Lower Baseline Limit attribute value to determine the number of standard deviations below the average that defines the lower baseline limit. If □ disabled: NNMi does not define the lower baseline limit.
Lower Baseline Limit	Enter the number of standard deviations below the average values that NNMi should use to determine the lower baseline limit.

Troubleshooting Monitoring Configuration

Determine Reasonable Threshold Settings

You must decide how to define normal behavior for devices in the associated Node Group or Interface Group. You can then set reasonable thresholds for the group and avoid Threshold Incident storms. See "Examples of Count-Based Threshold Monitoring" on page 364 and "Examples of Time-Based Threshold Monitoring" on page 368.

Create a Node Group or Interface Group filter that includes the devices you want to monitor. Export the Node Group or Interface Group filter to HP Network Node Manager iSPI Performance for Metrics Software. See "Creating Groups of Nodes or Interfaces" on page 294.

Enable Performance Monitoring for the Node Group or Interface Group. See "Node Group Settings for Monitoring" on page 391 or "Interface Group Settings for Monitoring (NNM iSPI Performance for Metrics)" on page 374. Then wait a minimum of 24 hours before following the steps below.

Access the NNM iSPI Performance for Metrics Headline report:

1. In the NNMi console, click **Actions** → **HP NNM iSPI Performance** → **Reporting - Report Menu**.
2. Click the link for **Headline**. The Headline report displays data from the past 24 hours from the time you request the report. So if you run the report at 5.03 p.m., the report includes data since 5.03 p.m. yesterday. Click the **Help** link in the report if you need information about how to use this report.
3. Open the **Topology Filters** panel and restrict your view to the network elements for which you are determining thresholds.
4. Click **Confirm Selection** to return to the report.
5. Open the **Time Controls** panel and select a start time and interval.

6. Click **Confirm Selection**.
7. The report appears using the filters you specified.
8. Study the Range & Exceptions graphs to guide your decision about what constitutes reasonable threshold settings. See online help for this report for information about how to read this report.

Find Threshold Results

The results of your threshold monitoring provide data in the following locations:

- NNM iSPI Performance for Metrics reports. See the HP Network Node Manager iSPI Performance for Metrics Software documentation.
- NNMi's **Monitoring** workspace → **Interface Performance** table view
- NNMi's **Node** form:
  - Node Components tab — Displays a list of node components associated with the selected node. Open threshold issues can influence the Status of node components.
  - Custom Polled Instances tab — Open Custom Poller threshold issues can influence results shown here.
  - Conclusions tab — Open threshold issues can influence conclusion calculations.
- NNMi's **Node Component** form:
  - Monitored Attributes tab — Displays a list of all related monitored attributes. The threshold results can influence the State of monitored attributes.
  - Conclusions tab— Open threshold issues can influence conclusion calculations.
- NNMi's **Interface** form:
  - Performance tab — Displays a list of currently configured thresholds related to the selected interface.

    > **Tip:** This information is also visible in the Monitoring workspace, Interface Performance view.

  - Conclusions tab — Open threshold issues can influence conclusion calculations.
- NNMi's **Layer 2 Connection Form** form:
  - Conclusions tab — Open threshold issues can influence conclusion calculations.

Confirm Threshold Configuration Settings

To view the threshold settings that produced the threshold state values above:

- Select an Interface, Node Component, or Node, click **Actions** → **Configuration Details** → **Monitoring Settings**, and then scroll down to the Count-Based Threshold Settings table and Time-Based Threshold Settings table.

> **Tip:** These tables do not appear if the selected Interface, Node Component, or Node is not a member of any Interface Group or Node Group with configured thresholds.

Threshold Monitoring Behavior After a System Restart or Configuration Change

After a network device is restarted, NNMi does the following:

- NNMi retains the device's former State value and updates the State value as soon as the new State is positively identified based on current configuration settings for Discovery and Monitoring.
- If the prior State value were **Not Polled**, NNMi changes the State to **Nominal** before determining the new State.

After a Threshold setting is re-configured, NNMi can positively identify the current device State when any the following criteria are met:

- For Count-Based Thresholds, High Trigger Count or Low Trigger Count is reached.
- For Time-Based Thresholds:
    - High Window Duration or Low Window Duration is reached.
    - NNMi receives enough data samples to identify the State. For example, if the Threshold definition setting is monitoring a value for 20 out of 30 minutes and the threshold is crossed within the first 20 minutes, then NNMi can update the State after 20 minutes has passed.

Monitor Router Redundancy Groups (*NNMi Advanced*)

NNMi monitors state and priority information for any discovered objects in the network. These objects include Router Redundancy Members and Tracked Objects. See Router Redundancy Group View for more information about Router Redundancy Groups and the objects associated with them.

The polling interval used is the Fault Polling Interval that is set for the node associated with the Router Redundancy Member or Tracked Object.

If you do not want these objects polled:

- Set the Management Mode for each node to **Not Managed** or **Out of Service**. See "Stop or Start Managing an Object" on page 456 for more information about Management Mode.
- Disable all Router Redundancy Group monitoring. See Set Global Monitoring.

NNMi Advanced also uses Router Redundancy Group objects when calculating a Path View between two nodes that have IPv4 addresses. See Path View with NNMi Advanced for more information.

Current Health of the State Poller Service

At any time, you can check the current health statistics about the State Poller Service.

To see a report of the health of the State Poller Service, click **Help** → **System Information** and navigate to the **State Poller** tab. For more information see Displaying NNMi System Information.

The State Poller Service contributes towards discovery and ongoing monitoring. See "About Each NNMi Service" on page 82.

Verify the Monitoring Settings

After the NNMi administrators configure the monitoring settings, configuration for particular objects can be verified to ensure that everything is working correctly. Examples of objects that have Monitoring Settings reports include Nodes, Interfaces, IP addresses, Router Redundancy Groups, Tracked Objects, and Node Components. Open the object's form and use the **Actions** → **Configuration Details** → **Monitoring Settings** menu item to display the report.

> **Tip:** You can right-click any object in a table or map view to access the **Actions** menu.

(*NNMi Advanced*) If the Global Network Management feature is enabled and you are signed into a Global Manager:

- Node managed by the Global Manager = **Actions** → **Configuration Details** → **Monitoring Settings** opens a report, provided by the Global Manager (NNMi management server).
- Node managed by a Regional Manager = **Actions** → **Configuration Details** → **Monitoring Settings** accesses that Regional Manager (NNMi management server) and requests the report.

  > **Note:** You must sign into that Regional Manager unless your network environment enables Single Sign-On (SSO) to that Regional Manager through the Global Manager. For more information, see the “Configuring Single Sign-On for Global Network Management” section in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.

To verify the monitoring configuration for a Node (SNMP Agent), Interface, IP address, or Card:

1. Navigate to the view for that object (for example, **Inventory** workspace, **Nodes)** view.
2. Select the row representing the object information.
3. Select **Actions** → **Configuration Details** → **Monitoring Settings**.

   NNMi displays the monitoring configuration settings for the selected object.

> **Note:** This menu item also is available on any object's form.

To verify the monitoring configuration for a Router Redundancy Member:

1. Navigate to a Router Redundancy Group view (for example, **Inventory** workspace, **Router Redundancy Groups** view).
2. Double-click the row representing the Router Redundancy Group configuration you want to see.
3. From the Router Redundancy Members tab, double-click the row representing the Router

Redundancy Member configuration you want to see.

4. Select **Actions** → **Configuration Details** → **Monitoring Settings**.

   NNMi displays the monitoring configuration settings for the selected object.

To verify the monitoring configuration for a Tracked Object:

1. Navigate to a Router Redundancy view (for example, **Inventory** workspace, **Router Redundancy Groups** view).
2. Double-click the row representing the Router Redundancy Group.
3. From the Router Redundancy Members tab, double-click the row representing the Router Redundancy Group Member.
4. From the Tracked Objects tab, double-click the row representing the Tracked Object.
5. Select **Actions** → **Configuration Details** → **Monitoring Settings**.

   NNMi displays the monitoring configuration settings for the selected object.

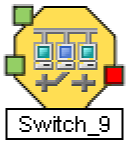
To verify the monitoring configuration for a Node Component:

1. Navigate to the view for that object (for example, **Inventory** workspace, **Nodes** view).
2. Double-click the row representing the Node Component Configuration.
3. Select the **Node Component** tab.
4. Double-click the row representing the object information.
5. Select **Actions** → **Configuration Details** → **Monitoring Settings**.

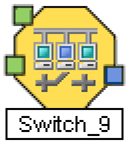
   NNMi displays the monitoring configuration settings for the selected object.

Check status and connectivity of important interfaces.

1. Open a Layer 2 Neighbor View map of each important interface's parent device. See Viewing Maps (Network Connectivity).
2. Each connected interface has a little square symbol around the edge of the parent device's map symbol. For example:

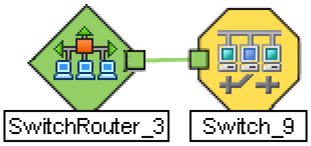


3. Hover your mouse over the square to verify the identify of your important interface on the map.
4. Verify that the status color of each important interface is not 🟦 Unknown or 🟧 **Unmanaged**[1] (see About Status Colors). For example:



[1] Indicates the Management Mode is "Not Managed" or "Out of Service".

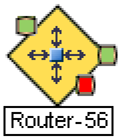
5. By default, NNMi only monitors the health of connected interfaces. A line appears on the map between interfaces when they are connected. For example:



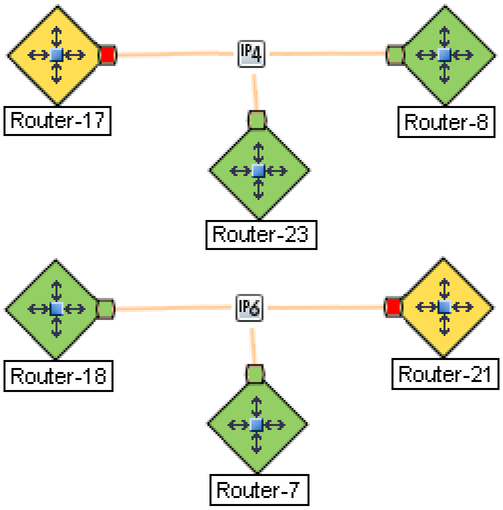
6. To add a connection, see "Add or Delete a Layer 2 Connection" on page 284.

Check status and connectivity of important addresses.

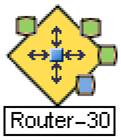
1. Open a Layer 3 Neighbor View map of each important parent device. See Viewing Maps (Network Connectivity).
2. Each address that is connected to another address in the same subnet has a little hexagon symbol around the edge of the parent device's map symbol. For example:



3. Hover your mouse over the hexagon to verify the identify of your important address on the map.
4. NNMi monitors the health of addresses only if you enable ICMP Address Monitoring. A line appears on the map between addresses when they are connected. The line represents the subnet. For example:



5. If ICMP Address Monitoring is enabled, verify that the status color of each important address is not 🟦 Unknown or 🟧 **Unmanaged**[1] (see About Status Colors). For example:



6. To add a connection, see "Add or Delete a Layer 2 Connection" on page 284.

[1] Indicates the Management Mode is "Not Managed" or "Out of Service".

See "Configure NNMi Monitoring Behavior" on page 340 for information about establishing monitoring behavior.

Monitor Status Distribution for Network Objects

NNMi enables you to view the overall health of your network by providing Stacked Area Graphs that display the distribution of Node, Interface, and IP Address Status information over time.

> **Tip:** If you do not want to display unpolled objects (No Status), use the **File** → **Select Area** menu option and clear the **No Status** check box.

To view Status Distribution Graphs:

1. Select **Tools** → **Status Distribution Graphs**.
2. Select the object type for which you want to display Status distribution. For example, **Node Status**.

   NNMi displays a Stacked Area Graph of the distribution of the object's Status over time.

   See **Help** → **Using Stacked Area Graphs** from the Graph menu bar for more information about using Stacked Area Graphs.

See "Configure NNMi Monitoring Behavior" on page 340 for information about establishing monitoring behavior.

Create Custom Polling Configurations

> **Tip:** Check to see if the threshold you want is already defined. See "About Threshold Settings Provided by NNMi" on page 354.

The Custom Poller feature enables you to take a proactive approach to network management by using SNMP MIB Expressions to specify additional information that NNMi should poll. You can also specify States that should be assigned to polled MIB Expression values, including any thresholds that should be set and monitored.

For example, if you have the `HOST-RESOURCES-MIB` loaded on your NNMi management server, you might want to monitor additional information using a single MIB variable, such as `hrDeviceStatus`, so that you can monitor information about a COM (communication) port, Loopback interface, or Ethernet Adapter Status. You might also want to monitor additional information using multiple MIB variables. For example, disk utilization could be calculated and polled using a MIB Expression similar to the following:(`hrStorageAllocationUnits` * `hrStorageSize`) / (`hrStorageUsed` * `hrStorageAllocation`)

Note the following:

- The MIB variables included in the MIB Expression that you want NNMi to poll must be loaded on the NNMi management server.
- A Custom Poller Policy is applied to the selected node or all the nodes in its specified Node Group as follows:

- At the time the Policy Active State attribute is set to **Active**. See "Create a Policy" on page 449 for more information.
- Each time the network is rediscovered as specified by the **Rediscovery Interval**. See "Adjust the Rediscovery Interval" on page 210 for more information.
- Each time you select **Actions** → **Polling** → **Configuration Poll** from the NNMi console.

As an Administrator, to configure Custom Polling you want to perform the following tasks:

1. *Prerequisite:* Install the MIB files needed for SNMP communication with the devices in your network environment:
   a. "Load MIBs" on page 1458
   b. "Unload MIBs " on page 1472
2. "Enable or Disable Custom Poller" below
3. "Create a Custom Poller Collection" on the next page
   a. "Configure Basic Settings for a Custom Poller Collection" on page 423
   b. "Specify the MIB Variable Information for a Custom Poller Collection" on page 429
   c. "Configure Threshold Information for a Custom Poller Collection" on page 441
   d. "Configure Comparison Maps for a Custom Poller Collection" on page 446
4. "Create a Policy" on page 449
5. "Create a Report Group (NNM iSPI Performance for Metrics)" on page 452

Refer to *HP Network Node Manager i Software Step-by-Step Guide to Custom Poller* white paper - available at: `http://h20230.www2.hp.com/selfsolve/manuals` for more details about configuring `Custom Poller`.

Enable or Disable Custom Poller

The Custom Poller Configuration form enables you to enable or disable your Custom Poller Collections. You can also view the Custom Poller Collections and Policies that have been created.

> **Note:** Custom Poller is not enabled by default. When Custom Polling is disabled, the State of Polled Instances retain the most recent value before Custom Poller was disabled.

To enable Custom Poller:

1. Navigate to the **Configuration** workspace.
2. Expand the **Monitoring** folder.
3. Select **Custom Poller Configuration**.
4. Click **Enable Custom Poller** ☑.
5. Click the **Save** icon.

   To verify that Custom Poller is working as expected, see the report on the Custom Poller tab in **Help** → **System Information**.

To disable Custom Poller:

1. Navigate to the **Configuration** workspace.
2. Expand the **Monitoring** folder.
3. Select **Custom Poller Configuration**.
4. Click to clear **Enable Custom Poller** □.
5. Click the **Save** icon.

   To verify that Custom Poller is working as expected, see the report on the Custom Poller tab in **Help** → **System Information**.

The Custom Poller Collections tab enables you to create a Custom Poller Collection. See "Create a Custom Poller Collection" below for more information.

The Policies tab enables you to create one or more policies for a Collection. See "Create a Policy" on page 449 for more information.

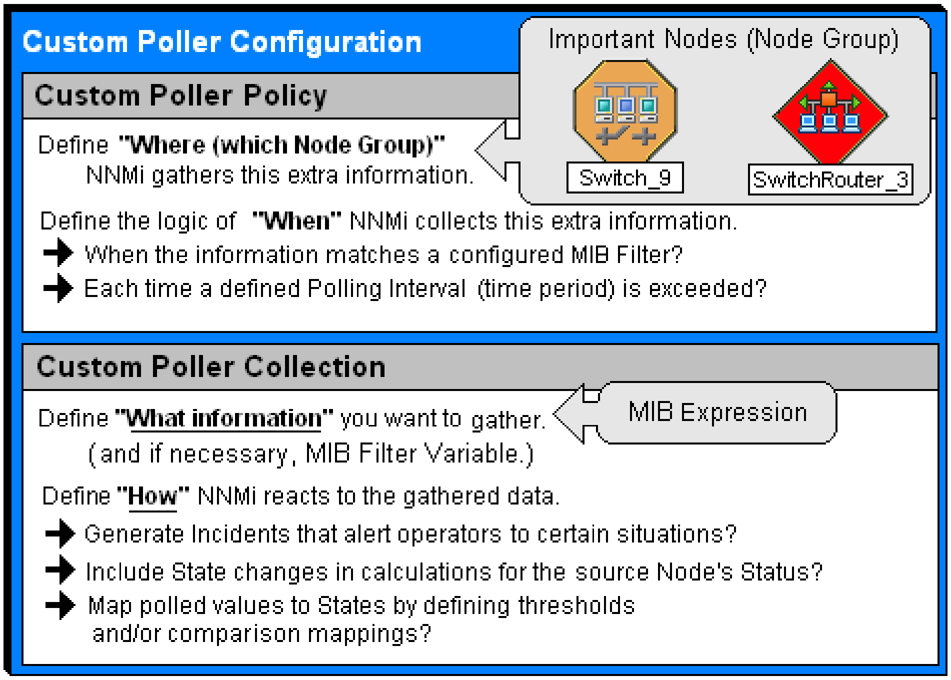
Create a Custom Poller Collection

A Custom Poller Collection defines the information you want to gather (poll) as well as how NNMi reacts to the gathered data. For example, you can specify whether you want to do either of the following:

- Configure Thresholds or Comparison Maps that map polled MIB Expression values to States and optionally causes incidents to be generated.
- Include State changes in calculations for the source Node's Status.

Each Custom Poller Collection can have one or more Policies. Each Policy specifies the Node Group from which you want to gather the additional information. The first time a MIB Expression is validated with discovery information, the results appear in a Polled Instance object. The Polled Instance object is updated whenever a change in State occurs and includes the most recent polled value that caused the State to change.

Click here for a diagram that describes Custom Poller Collections and their associated Policies:



Custom Poller Configuration

Custom Poller Policy

Define **"Where (which Node Group)"** NNMi gathers this extra information.

Define the logic of **"When"** NNMi collects this extra information.

- When the information matches a configured MIB Filter?
- Each time a defined Polling Interval (time period) is exceeded?

Custom Poller Collection

Define **"What information"** you want to gather. (and if necessary, MIB Filter Variable.)

Define **"How"** NNMi reacts to the gathered data.

- Generate Incidents that alert operators to certain situations?
- Include State changes in calculations for the source Node's Status?
- Map polled values to States by defining thresholds and/or comparison mappings?

To create a Custom Poller Collection, do the following:

1. Navigate to the **Custom Poller Collections** tab.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Expand the **Monitoring** folder.
   c. Select the **Custom Poller Configuration** form.
   d. Select the **Custom Poller Collections** tab.
   e. Do one of the following:
      - To create a Custom Poller Collection, click the New icon.
      - To edit a Custom Poller Collection, double-click the row representing the configuration you want to edit.
2. Make your configuration choices (see table).
3. Click **Save and Close**.

> **Note:** When you save a Collection configuration, each Policy for that Collection changes to Active State **Suspended**. When you are finished making your Custom Poller Configuration changes, set the Active State to **Active** for each of the policies in the Custom Poller Collection that you want to be in use. To make a Policy active, access the Custom Poller Configuration: Policy tab, open each associated Policy, and change the Active State to **Active**. See "Create a Policy" on page 449 for more information.

To verify that Custom Poller is working as expected, see the report on the Custom Poller tab in **Help → System Information**.

Custom Poller Collection Configuration Tasks

Task	How
"Configure Basic Settings for a Custom Poller Collection" below	Provide the basic information for a Custom Poller Collection configuration.
"Specify the MIB Variable Information for a Custom Poller Collection" on page 429	You specify the MIB Expression you want to poll. Use the MIB Expression editor to specify the MIB Variable and any constant or arithmetic operator you want to use in the MIB Expression. Navigate the MIB tree to select each MIB Variable.
"Configure Threshold Information for a Custom Poller Collection" on page 441	*Optional*. Specify minimum and maximum threshold values for the MIB Expression results and assign these thresholds to States.
"Configure Comparison Maps for a Custom Poller Collection" on page 446	*Optional*. Use Comparison Maps to assign a State value to a potential polled value of a MIB Expression.

Note: Thresholds and Comparison Maps contribute to State calculations. If you configure both Thresholds and Comparison Maps, NNMi first checks Threshold settings to determine State values. If the Threshold evaluates to non-Normal, NNMi uses the Threshold settings to determine State values. If the Threshold evaluates to Normal, NNMi checks for a non-Normal State using any Comparison Maps configuration.

Configure Basic Settings for a Custom Poller Collection

The Basic settings for a Custom Poller Collection include the Name of the Custom Poller Collection as well as whether to have this Collection affect a Node's Status or generate incidents under specified conditions. You also use the Basic settings to configure whether NNMi exports Custom Poller Collection metrics to a comma-separated values (CSV) file for use in other applications.

To configure the Basic settings for a Custom Poller Collection:

1. Navigate to the **Custom Poller Collection** form.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Expand the **Monitoring** folder.
   c. Select the **Custom Poller Configuration** view.
   d. Select the **Custom Poller Collections** tab.

   e. Do one of the following:

      - To create a Collection, click the ✱ New icon.
      - To edit a Collection, double-click the row representing the configuration you want to edit.

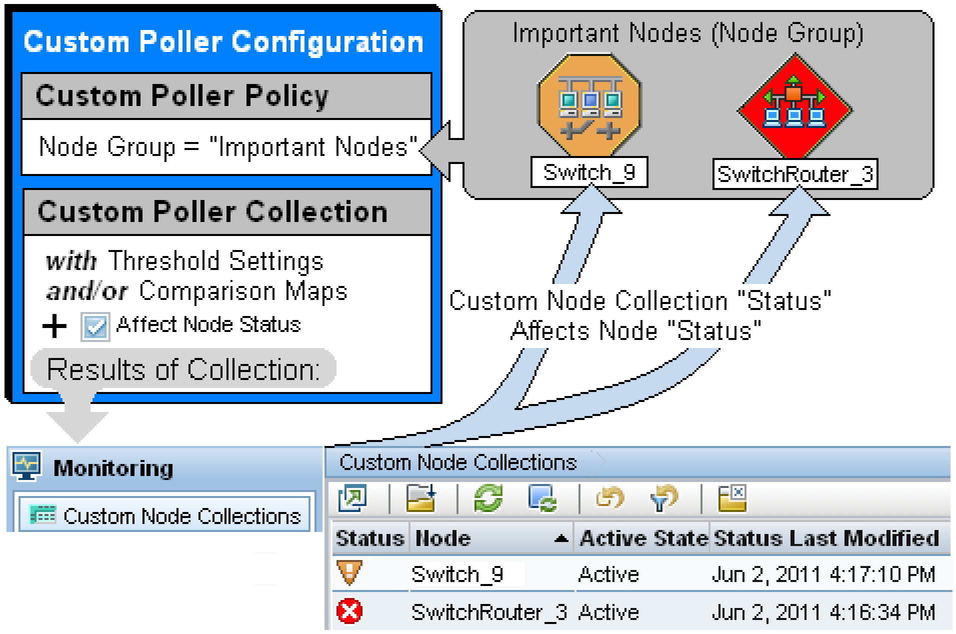
2. Provide the required basic settings (see the Basics for this Custom Poller Collection table).
3. Complete the configuration for this Custom Poller Collection configuration, if you have not already done so:
4. Click **Save and Close** to return to the **Custom Poller Configuration** form.

   To verify that Custom Poller is working as expected, see the report on the Custom Poller tab in **Help** → **System Information**.

Basics for this Custom Poller Collection

Attribute	Description
Name	The name for the Custom Poller Collection configuration. The name can be up to 50 alphanumeric characters. Spaces are permitted. The following special characters (<, >, ", ', &, \, \, #) are not permitted. **Note:** The Custom Poller Collection name appears in any incidents generated as a result of the collection. Specify a name that will help you to identify the MIB information being polled.
Affect Node Status	If enabled ☑, NNMi uses the Status of a Custom Node Collection to affect the associated topology Node's Status. To understand how the topology Node Status is affected, you must understand the relationship between a Custom Node Collection and a topology Node. Click here for more information: As shown in the following diagram, a Custom Node Collection identifies each topology node that has at least one associated Custom Poller Collection and Custom Poller Policy pair.

Basics for this Custom Poller Collection, continued

<table>
<tr><th>Attribute</th><th>Description</th></tr>
<tr><td></td><td>


<table>
<tr><th>Status</th><th>Node</th><th>Active State</th><th>Status Last Modified</th></tr>
<tr><td>▽</td><td>Switch_9</td><td>Active</td><td>Jun 2, 2011 4:17:10 PM</td></tr>
<tr><td>☒</td><td>SwitchRouter_3</td><td>Active</td><td>Jun 2, 2011 4:16:34 PM</td></tr>
</table>
</td></tr>
<tr><td>Generate Incident</td><td>

If enabled ☑, NNMi generates an incident when a Threshold (defined on the Thresholds tab) is reached or exceeded, or when a specified MIB returns a value that causes the Node's *State* to be other than **Normal** (defined on the Comparison Maps tab).

To generate incidents for the Custom Node Collection, select **Custom Node Collection**.

To generate incidents for Custom Polled Instances, select **Custom Polled Instance**.

If disabled □, NNMi does not generate any incidents for Custom Node Collections or Custom Polled Instances.

To understand how Custom Node Collection incidents are generated, it is important to understand the relationship between a Custom Poller Policy and a Custom Poller Collection. Click here for more information:

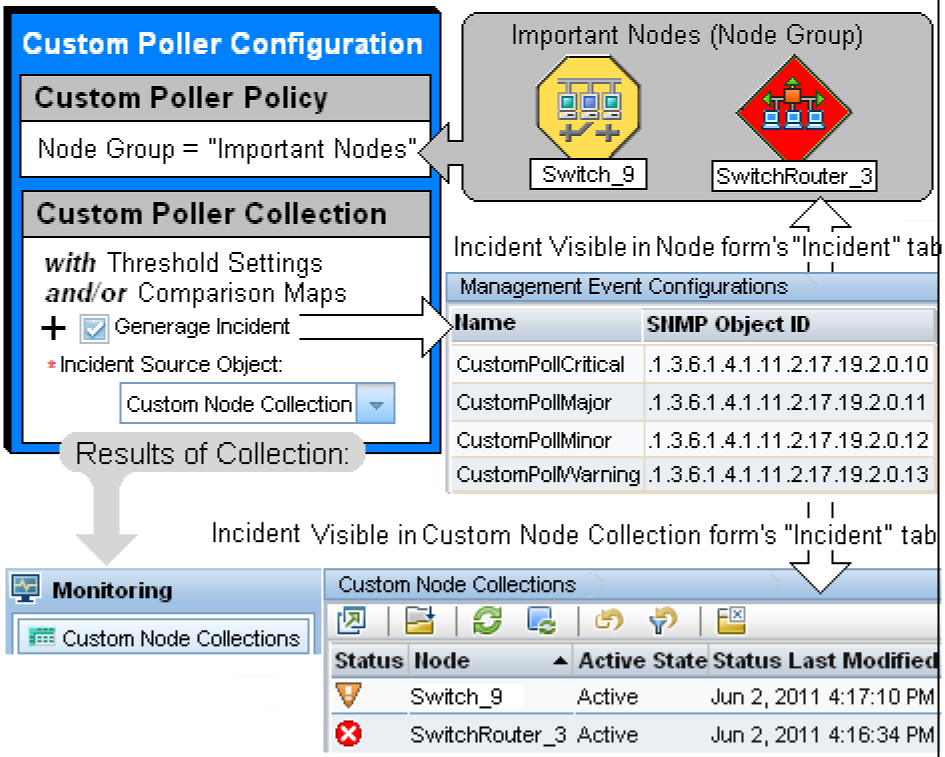
- Each Custom Poller Collection is associated with a Custom Poller Policy that identifies the Node Group to which the policy and collection apply.
- The results of each Custom Poller Collection and Custom Poller Policy pair appear in one row of the **Monitoring** workspace's Custom Node Collection view. A Custom Node Collection identifies each topology node that has at least one associated Custom Poller Collection and Custom Poller Policy pair.

  Multiple Custom Poller Collection and Custom Poller Policy pairs can be associated with the same Node Group. Results appear as multiple rows for each Node Group member in the Custom Node Collection view.
- Click here to view a diagram of this relationship.

</td></tr>
</table>

Basics for this Custom Poller Collection, continued

<table>
<tr><th>Attribute</th><th>Description</th></tr>
<tr><td></td><td>



<table>
<tr><th>Name</th><th>SNMP Object ID</th></tr>
<tr><td>CustomPollCritical</td><td>.1.3.6.1.4.1.11.2.17.19.2.0.10</td></tr>
<tr><td>CustomPollMajor</td><td>.1.3.6.1.4.1.11.2.17.19.2.0.11</td></tr>
<tr><td>CustomPollMinor</td><td>.1.3.6.1.4.1.11.2.17.19.2.0.12</td></tr>
<tr><td>CustomPollWarning</td><td>.1.3.6.1.4.1.11.2.17.19.2.0.13</td></tr>
</table>

<table>
<tr><th>Status</th><th>Node</th><th>Active State</th><th>Status Last Modified</th></tr>
<tr><td>▽</td><td>Switch_9</td><td>Active</td><td>Jun 2, 2011 4:17:10 PM</td></tr>
<tr><td>⊗</td><td>SwitchRouter_3</td><td>Active</td><td>Jun 2, 2011 4:16:34 PM</td></tr>
</table>

When generating incidents for Custom Node Collections, note the following:

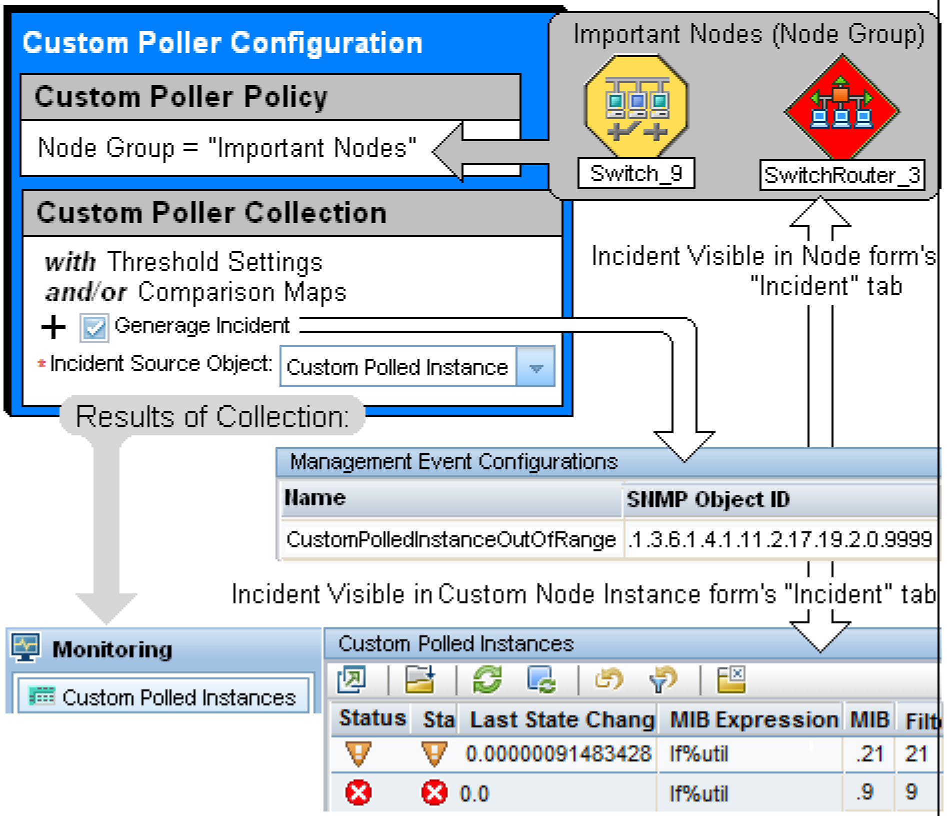
- If a Custom Node Collection meets or exceeds a configured threshold, an incident is generated for the associated Custom Node Collection.
- NNMi generates only one incident per Custom Node Collection. This means if multiple instances within the Custom Node Collection have a Status other than Normal, NNMi generates only one incident using the details for the highest severity instance.
- If multiple instances within the Custom Node Collection have a Status other than Normal and more than one of them has the highest severity, NNMi selects one of the Custom Node Collection instances to generate the incident.
- The most severe incident status is then propagated from the Custom Node Collection to the corresponding node object.
- If the Custom Node Collection with the most severe status returns to normal, NNMi closes the corresponding incident. If another instance in the Custom Poller Collection has a status other than normal, NNMi generates a new incident using the next highest severity.

To understand how Custom Polled Instance incidents are generated, it is important to understand how Custom Polled Instances are generated. Click here for more information:

- The first time a MIB Expression is validated with discovery information, the results appear in a Custom Polled Instance object. A Custom Polled Instance

</td></tr>
</table>

Basics for this Custom Poller Collection, continued

Attribute	Description
	represents the results of a MIB expression when it is evaluated against a node. The Custom Polled Instance is updated whenever a change in State occurs and includes the most recent polled value that caused the State to change. • A node can be associated with multiple Custom Polled Instances when its associated MIB expression includes MIBs that have multiple instances per node. For example, the associated MIB expression might perform a calculation using the ifInOctets and ifOutOctets MIB values. Using the MIB Filter and MIB Filter Variable specified, NNMi calculates these values for each interface that meets the filter criteria and that is associated with a node in the Custom Poller Collection. • Click here to view a diagram of this relationship.  Custom Poller Configuration Custom Poller Policy Node Group = "Important Nodes" Custom Poller Collection with Threshold Settings and/or Comparison Maps Generage Incident Incident Source Object: Custom Polled Instance Results of Collection: Important Nodes (Node Group) Switch_9 SwitchRouter_3 Incident Visible in Node form's "Incident" tab Management Event Configurations Name SNMP Object ID CustomPolledInstanceOutOfRange .1.3.6.1.4.1.11.2.17.19.2.0.9999 Incident Visible in Custom Node Instance form's "Incident" tab Monitoring Custom Polled Instances Custom Polled Instances Status Sta Last State Chang MIB Expression MIB Filt 0.00000091483428 If%util .21 21 0.0 If%util .9 9 When generating incidents for Custom Polled Instances, note the following: • The Source Object is the Custom Polled Instance and the Source Node is the Node associated with the specified Custom Node Collection. • If the Status of the Custom Polled Instance incident changes from Critical to Major, Minor, or Warning, NNMi cancels the Critical incident and replaces it with the incident that has a Status of Major, Minor, or Warning • If the Status of the Custom Polled Instance Incident changes from Major, Minor, or Warning to Critical, the current incident is canceled and replaced by the

Basics for this Custom Poller Collection, continued

Attribute	Description
	incident that has a Critical Status. • When the Status of the Custom Polled Instance changes to Normal, the Custom Polled Incident is Closed. • If a Custom Poller Policy's Active State becomes Inactive, NNMi deletes any Custom Polled Instances associated with the Custom Poller Policy and Closes any associated Custom Polled Instance incidents.

Export Custom Poller Collection	If enabled ☑, NNMi exports the Custom Poller Collection to a comma-separated values (CSV) file that is written to the following directory:
	Windows:
	`%NnmDataDir%\shared\nnm\databases\custompoller\export\final`
	Unix:
	`$NnmDataDir/shared/nnm/databases/custompoller/export/final`
	When exporting Custom Poller Collections, note the following:
	• NNMi includes the following information in the CSV file:
	▪ Node UUID
	▪ IP address
	▪ Node Name (Host Name of the Node)
	▪ The MIB expression or the numeric Object Identifier of the MIB variable
	▪ Time stamp (in milliseconds)
	▪ Poll interval (in milliseconds)
	▪ MIB Instance (number used to identify the row in the MIB table)
	▪ Metric value
	▪ Display Attribute (See "MIB Expressions Form (Custom Poller)" on page 431for more information)
	▪ Filter Value (See "MIB Expressions Form (Custom Poller)" on page 431 for more information)
	• By default, NNMi accumulates the data and writes the metrics to the CSV file, one metric per Custom Poller Collection instance, every 5 minutes.
	• NNMi names each CSV file using the Custom Poller Collection name, appended with the timestamp (yyyymmddHHmmssSSS).
	• NNMi monitors the `custompoller` directory to ensure that the Custom Poller metrics do not fill the disk. By default, after the custompoller directory has consumed more than one gigabyte of disk space, NNMi removes the oldest metric files as it writes new files to the disk.

Basics for this Custom Poller Collection, continued

Attribute	Description
	• See the HP Network Node Manager i Software Deployment Reference for information about how to change default values, including the directory name, disk size, and the interval at which NNMi accumulates the data before writing the metric files to the disk. • If you have a High Availability (HA) environment, NNMi places the CSV files on the shared disk. • If you are using Application Failover, NNMi replicates these files to the failover system. See the HP Network Node Manager i Software Deployment Referencefor more information about HA and Application Failover. • If you change the name of a Custom Poller Collection or the MIB Expression associated with a Custom Poller Collection that is exported, NNMi removes all of the historical data for that Custom Poller Collection. If disabled □, NNMi does not export the Custom Poller Collection information.
Compress Export File	If enabled ☑, NNMi exports the Custom Poller Collection in compressed format and appends `*.gz` to the `*.csv` file suffix. If you have more than one Custom Poller Collection with the same name, note the following: • If at least one of those Custom Poller Collections has Compress Export File enabled, NNMi compresses all of the exported Custom Poller Collections with the same name. • NNMi writes the Custom Poller Collection information to the same CSV file. If disabled □, NNMi does not compress the CSV file.

See "Specify the MIB Variable Information for a Custom Poller Collection" below for information about the Variable attributes.

See "Configure Threshold Information for a Custom Poller Collection" on page 441 for information about configuring Thresholds.

Specify the MIB Variable Information for a Custom Poller Collection

When specifying the MIB variable information, note the following:

- Each MIB Variable included in the MIB Expression must be loaded on the NNMi management server.
- You specify only one MIB Expression per Custom Poller Collection.
- You navigate the MIB tree to select a MIB Variable to include in a MIB Expression.

Variable Attributes

Attribute	Description
MIB Expression	You create a MIB Expression by using the MIB Expression form. If your NNMi Security configuration permits, to access the MIB Expression form, click the Lookup icon and do one of the following: • Select **Quick Find** to select an existing MIB expression. • Select **Open** to edit the current MIB expression. • Select **New** to create a MIB expression. See "MIB Expressions Form (Custom Poller)" on the next page for more information.
MIB Filter Variable	The MIB Filter Variable is the MIB variable value you want to use as a filter to determine which instances of the MIB expression to Custom Poll. If you specify a MIB Filter Variable, you must also specify a MIB Filter (value). For example, because a node can have multiple interfaces, MIB expressions containing interface information have multiple instances and require you to use a MIB Filter Variable and MIB Filter (value) to specify which interfaces you want NNMi to poll. You might use a MIB Filter Variable of `ifIndex` and a MIB Filter (value) of `1`. In this example, NNMi creates a Polled Instance for each interface with an (interface index) ifIndex value of 1 in the Node Group or Interface Group specified by the associated Custom Poller Policy. See "Create a Policy" for more information about Custom Poller Policies. Valid types for MIB Filter Variables include the following: • Integer • Unsigned Integer • Gauge • Octet String • IpAddress (IPv4 only) **Note:** The MIB Filter Variable must also be a MIB variable that has multiple instances (Table Entry MIB). Click the icon to open the MIB tree and select the MIB variable you want to use. When using MIB Filter Variables, note the following: • If you do not see a MIB that you recently loaded, close the Custom Poller Collection form, wait 1 minute for NNMi to cache the new MIB information, then open the MIB tree again. • To remove an unwanted MIB Filter Variable: a. Delete any MIB Filter Values from all Policies associated with the Custom

Variable Attributes, continued

Attribute	Description
	Poller Collection. b. Edit the Custom Poller Collection to remove the MIB Filter Variable.

To specify Threshold information for the Custom Poller Collection, see "Configure Threshold Information for a Custom Poller Collection" on page 441

MIB Expressions Form (Custom Poller)

You can access the MIB Expression form in the following ways:

- From the **Configuration** workspace > **MIBs**folder > **MIB Expressions** view.
- From the **Configuration** workspace > **Monitoring** folder > **Custom Poller Configuration** form
- From the **MIB Specification** form. (Used when configuring SNMP Graph actions.)

When you want to create a MIB Expression to be used in Graphs, use the **MIB Expressions** view. See "MIB Expression Form (Line Graph)" on page 1474 for more information.

When you want to create a MIB Expression to be used in a Custom Poll, use the **Custom Poller Configuration** form.

Note: You can re-use any MIB Expression that you create for NNMi graphs or for Custom Poller. Use "MIB Expressions View" on page 1474 to see a list of the available MIB Expressions. Use the "Loaded MIBs View" on page 1451 to see a list of the MIBs loaded on the NNMi management server.

To create a MIB Expression using the Custom Poller Configuration form:

1. Navigate to the **Custom Poller Configuration** form.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Expand the **Monitoring** folder.
   c. Select the **Custom Poller Configuration** form.
   d. Select the **Custom Poller Collections** tab.
   e. Do one of the following:
      - To create a Collection, click the **New** icon.
      - To edit a Collection, double-click the row representing the configuration you want to edit.
2. In the MIB Expression attribute, click the **Lookup** icon and do one of the following:
   - Select **Quick Find** to select and edit an existing MIB expression.
   - Select **Open** to edit the current MIB expression.

- Select **New** to create a MIB expression.

3. Provide the required basic settings (see the MIB Expression Basic Attributes table).

4. Click **Save and Close** to return to the **Custom Poller Configuration** form.

   You must save the MIB Expression before you use **Actions** → **Graph MIB Expression**.

5. To test your MIB Expression, select **Actions** → **Graph MIB Expression**. See "Test a MIB Expression (Custom Poller)" on page 436 for more information.

   Tip: You can right-click any object in a table or map view to access the **Actions** menu.

   The NNMi administrator determines the label that is used to identify the data instances that are displayed in Line Graphs using the Instance Display Configuration (see the Instance Display Configuration table). If the Instance Display Configuration is not set, NNMi identifies each instance that appears in a Line Graph using the Node's short DNS Name followed by the MIB Instance value in the format: <*node_name*> -<*MIB_instance_value*>. This value also appears as the Display Attribute in the Custom Polled Instance View.

MIB Expression Basic Attributes

Attribute	Description
Unique Key	Used as a unique identifier when exporting and importing MIB Expression definitions. To ensure that the value you enter is unique, it is recommended that you use the Java name space convention when providing this value. It is also useful to include the label value as part of the unique key as shown in the following example: `com.<your_company_name>.nnm.mibexp.<mib_expression_name>` The maximum length is 80 characters. **Note:** Unlike the Unique Key attributes associated with other objects, you can change the MIB Expression configuration's Unique Key value at any time.
Name	The name you want to use for the MIB information being polled. This name can be the same name as a MIB Variable used in the MIB Expression, or you can enter a name of your choice. Type a maximum of 50 characters. Alpha-numeric and special characters (~ ! @ $ % ^ * ( ) _+) are permitted. No spaces are permitted.
Author	Indicates who created or last modified the MIB Expression. **Caution:** If the Author attribute value is **HP Network Node Manager**, any changes are at risk of being overwritten in the future. • Click **Lookup** and select **Show Analysis** to display details about the currently selected Author. • Click **Quick Find** to access the list of existing Author values.

MIB Expression Basic Attributes, continued

Attribute	Description
	• Click ✱ **New** to create an Author value.
Expression	Click the ▤ button to access the MIB Expression editor. See "Use the MIB Expression Editor (Custom Poller)" on page 437 for information about using the MIB Expression editor. Valid types for the MIB variables that can be included in a MIB expression for Custom Poller include the following: • Integer • Unsigned Integer • Gauge • Counter • Counter64 • TimeTicks • Octet String Note the following: • The MIB containing the variable must be loaded on the NNMi management server. • If a MIB Expression includes more than one MIB Variable that has multiple instances (Table Entry MIB), select a MIB Filter and MIB Filter Variable that can be consistently applied to each Table Entry MIB in the expression. • Although it is strongly discouraged, to configure Custom Polling for all instances of a repeating MIB, you can use the same MIB variable for both the MIB Expression and the MIB Filter Variable. • If your MIB Expression contains an invalid MIB Variable, NNMi is not able to create an associated Polled Instance. If Polled Instances are not created as expected, check the Custom Node Collection view for Discovery State and Discovery State Information values. • If Polled Instances are created, but errors occur while processing the MIB Expression data from a device's SNMP Agent, information is logged to the analysis.0.0.log file. Examples of possible errors include divide by zero (0) or data unavailable. See "Verify that NNMi Services are Running" on page 85for more information about log files. • When evaluating MIB expressions that include MIB variables of type Counter, Counter64 or Time_Ticks, NNMi evaluates the MIB Variable using the difference in value between the most recent poll and the poll before it. If you want NNMi to calculate a rate over time in seconds, divide the MIB Expression by sysUptime. For example: `(((ifInOctets+ifOutOctets)*8/ifSpeed)*100)/sysUpTime*0.01`

MIB Expression Basic Attributes, continued

Attribute	Description
	Tip: The sysUpTime variable is a value of hundredths of a second. When you want the rate in seconds, use `sysUpTime*0.01` in the MIB expression as shown in the previous example. - If you use a MIB variable of type Counter, Counter64 or Time_Ticks in the MIB Expression, NNMi automatically collects sysUpTime values if sysUpTime is not already in the MIB Expression. NNMi uses the sysUptime value to detect a system reboot. Any time a system reboot is detected, NNMi cannot determine the difference in values between polls for any Counter MIB variable and therefore does not calculate the MIB Expression for that poll.
Display numeric MIB OIDs in the Expression	Enables you to display the MIB object identifier (OID) rather than the MIB variable name in the MIB Expression. Select **Display MIB OIDs in the Expression** ☑ to replace any MIB variable name with the MIB OID value in the MIB Expression. Clear **Display MIB OIDs in the Expression** □ to display the MIB variable names rather than the MIB OIDs within the MIB Expression.
Description	NNMi provides the Description attribute to help you further identify the current MIB Expression configuration. Use the description field to provide additional information that you would like to store about the current MIB expression configuration. Type a maximum of 2000 characters. Alpha-numeric and special characters (~ ! @ # $ % ^ & * ( ) _+ -) are permitted.

Instance Display Configuration

Attribute	Description
Conversion Algorithm	Used to determine the format in which the value contained in the Display Variable appears in the NNMi console. **Note:** NNMi applies the Display Filter to each Display Variable to determine the value to display. Possible Conversion Algorithms are: - **Numeric** - Use this option to display the instance number returned by the SNMP query. This format is useful when no meaningful name is available in the MIB. For example, you might use this format to display CPU information. - **MIB Variable** - Use this option to display the value that is stored in the MIB variable you specify. To obtain each MIB variable value, NNMi appends the instance number to the MIB variable specified. The result from the SNMP query is converted to a text string and displayed.

Instance Display Configuration, continued

Attribute	Description
	• **Alphabetic** - Use this option to display information for legacy Cisco Arrow Point load balancers. When using this algorithm, each instance number returned by the SNMP query is treated as a set of ASCII characters instead of numbers. For example, the instance 101.120.97.109.112.108.101 would be displayed as 'example'. • **Interface Name** - Use this option to display the interface name. **Note:** The Interface Name option is only valid when an IfIndex value is returned as the instance number. The ifIndex value is then used to determine the Interface Name value. • **Interface Name Indirect** - Use this option to display the Interface Name value obtained from an indirect reference in the MIB table. For example, if the MIB variable you specify resides in an RMON MIB table, use this algorithm. **Note:** The **Interface Name Indirect** option is only valid when an OID is returned from an SNMP query that, when queried, returns an ifIndex value. The ifIndex value is then used to determine the Interface Name value using the "Interface Name" algorithm.
Display Variable	Select the MIB variable you want to display. NNMi uses the Conversion Algorithm you specify to determine how to obtain the Display Variable's value.
Display Filter	The value that NNMi displays for the Display Variable is determined by the criteria you provide here. This value is indicated as **Display Attribute** in the NNMi console. Enter a valid regular expression that specifies the pattern you want NNMi to match when determining the values to display. **Note:** NNMi uses the syntax defined for java regular expressions (java.util.regex Pattern class). NNMi finds the first character sequence that matches the Display Filter expression. If NNMi does not find a match for the Display Filter, it returns the Display Variable name. For example, if you have several interfaces with an ifDescr set to "FastEthernet" followed by a unique set of numbers for each interface (such as FastEthernet0/1, FastEthernet0/2, FastEthernet0/3, and so on), you can use the following Display Filter to display "Ethernet" followed by the unique set of numbers: `(Ethernet.*[0-9]+){1}` In the example, the following matches occur:

Instance Display Configuration, continued

Attribute	Description
	• `Ethernet` matches Ethernet • The `.*` matches 0/ • The `[0-9]+` matches any sequence of numbers • The `{1}` specifies to match the expression exactly one time In this example, possible Display Values include **FastEthernet0/1**, **FastEthernet0/2**, and **FastEthernet0/3**.

Test a MIB Expression (Custom Poller)

The Actions menu enables you to test the results of a MIB Expression using a Line Graph.

You must save the MIB Expression before you use **Actions** → **Graph MIB Expression**.

> **Tip:** You can right-click any object in a table or map view to access the **Actions** menu.

To graph the results for a MIB Expression:

1. Navigate to the **MIB Expression** form.

   a. From the workspace navigation panel, select the **Configuration** workspace.

   b. Expand the **MIBs** folder.

   c. Select the **MIB Expressions** view.

   > **Note:** You can also access the MIB Expressions form when creating Line Graphs and when creating Custom Poller Collections. See "MIB Expression Form (Line Graph)" on page 1474 and "MIB Expressions Form (Custom Poller)" on page 431 for more information.

2. Select the row representing the MIB Expression you want to graph.

3. Select **Actions** → **Graph MIB Expression**.

   The dialog for selecting a node apears.

4. Click the **Lookup** icon and select **Quick Find**.

5. Select the node you want to use to test your MIB Expression results.

   NNMi displays a Line Graph using the selected node and calculating the results for the MIB Expression you selected.

   Note the following:

   - *Line Graphs Only*. When evaluating MIB Expressions that include MIB variables of type Counter or Counter64, NNMi requests both the high capacity and low capacity counter variable for any interface instance. If the high capacity Counter64 is enabled for any given interface instance, NNMi uses the high capacity counter.

- *Custom Poller Only*. When evaluating MIB Expressions that include MIB variables of type Counter, NNMi requests only the low capacity counter information for any interface instance.

Use the MIB Expression Editor (Custom Poller)

Use the MIB Expression Editor to specify MIB Variables and any Constant values or arithmetic operators in your MIB Expression.

- For a description of each MIB Expression Editor option, see the table below.
- Before you start, review the MIB Expression Editor guidelines, click here.
  - As a general guideline, begin by writing out the MIB Expression. Then in the MIB Expression Editor, begin creating your MIB Expression by selecting your arithmetic operators (+, -, *, or /) from the outermost parenthesis to the innermost parenthesis. Each time you specify an arithmetic operator (+, -, *, or /), NNMi creates a set of parenthesis to specify the ordering of the mathematical calculation.
  - When adding arithmetic operators (+, -, *, or /) to a MIB Expression, first click to select the location in the MIB Expression at which you want to add the arithmetic operator.
  - Click to select the arithmetic operator (for example +) in the MIB Expression, before selecting the MIB variable or Constant value that you want to add, subtract, multiply or divide.
  - NNMi inserts arithmetic operators, MIB Expressions, and Constant values from the left to right.
  - To replace an arithmetic operator use the <> (Change Operator) button (see table).
  - To replace a MIB Variable or Constant value, click to select the existing value in the MIB Expression and then select the new MIB variable or enter the new Constant value.

    > **Note:** You can replace a MIB Variable with another MIB Variable or with a Constant value. You can replace a Constant value with a MIB Variable or Constant value.

  - You can drag any of the following items to a new location in the MIB Expression:
    - MIB variable
    - Constant value
    - An operation, such as **(IfInOctets + IfOutOctets)**
- For information about moving items to a new location within your MIB Expression, click here.
  - To move an arithmetic operation (for example, **(IfInOctets + IfOutOctets)**), click the arithmetic operator before dragging it to a new location.
  - To move a MIB Variable or Constant Value, click the MIB Variable or Constant Value you want to move before dragging it to a new location.
  - If you are moving the selected item to the right, NNMi places the item to the right of the new location.
  - If you are moving the selected item to the left, NNMi places the item to the left of the new location.

- As you drag a selected item, an underline indicates the current target location.
- If you drag a selected item past the outermost parenthesis, it is deleted. If desired, you can re-enter the value in the new location.

MIB Expression Example

To poll or graph a MIB Expression that calculates the percentage of available bandwidth on a half-duplex interface, create the following MIB Expression:

((((ifInOctets + ifOutOctets) * 8) / ifSpeed) * 100)

For an animated demonstration of creating this MIB Expression, click here.

For step-by-step instructions about creating this MIB Expression, click here.

To create the expression above, begin by specifying each arithmetic operator from the outermost parenthesis to the innermost parenthesis.

1. Click *. (multiply).
2. Click / (divide).

   Now that you have multiple entries in your MIB Expression, click to select the location in the MIB Expression to which you want to add each remaining arithmetic operators.
3. In the MIB Expression, click **/** (divide).

   The divide (/) arithmetic operator and its surrounding parenthesis should appear highlighted. Because NNMi inserts arithmetic operators, MIB variables, and Constant values from left to right, selecting / (divide) places the next arithmetic operator to the left of the divide arithmetic operator.
4. Click * (multiply).

   The multiply (*) arithmetic operator and its parenthesis should appear to the left of the divide arithmetic operator you previously selected.
5. In the MIB Expression, click the leftmost ***** (multiply).

   The multiply (*) arithmetic operator and its surrounding parenthesis should appear highlighted.
6. Click + (add).

   The add (+) arithmetic operator and its parenthesis should appear to the left of the multiply (*) arithmetic operator you previously selected.

   Now that you have specified the arithmetic operators, you are ready to add the MIB variables and Constant values. Begin by selecting the arithmetic operator in the MIB Expression to which you will add MIB variables, Constant values, or both. We will begin with the leftmost arithmetic operation.

   > **Note:** As you add your MIB variables or Constant values, make sure you first select the corresponding arithmetic operator within the MIB Expression.
7. In the MIB Expression attribute, click + (add).
8. Select the IfInOctets MIB Variable:

a. Click to open the MIB Variable Tree.

b. Navigate to **ifInOctets**.

c. Select **ifInOctets**.

d. Click **OK**.

The ifInOctets MIB variable should appear to the left of the add (+) arithmetic operator.

9. Select the IfOutOctets MIB Variable:

a. Click to open the MIB Variable Tree.

b. Navigate to **ifOutOctets**.

c. Select **ifOutOctets**.

d. Click **OK**.

The ifOutOctets MIB variable should appear to the right of the add (+) arithmetic operator.

You are ready to specify the Constant value 8 that corresponds with the leftmost multiply (*) arithmetic operator.

10. Click the leftmost * multiply.

11. In the Constant attribute, enter 8 and click Enter.

The value 8 should appear to the right of the multiply (*) arithmetic operator that you previously selected.

12. In the MIB Expression, click divide (/).

13. Select the IfSpeed MIB Variable:

a. Click to open the MIB Variable Tree.

b. Navigate to ifSpeed.

c. Double-click ifSpeed.

d. Click **OK**.

The ifSpeed MIB Variable name should appear to the right of the divide (/) arithmetic operator you previously selected.

14. Click the rightmost * (multiply)

15. In the Constant attribute, enter 100 and then click Enter.

16. The Constant value 100 should appear to the right of the divide (/) arithmetic operator you previously selected.

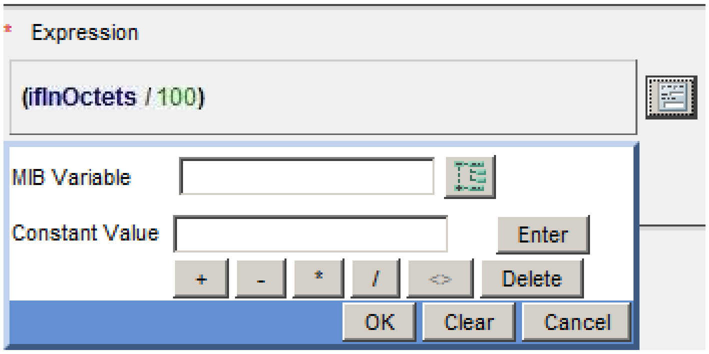
17. Click **OK** to save your MIB Expression.

The following table describes each of the MIB Expression Editor options.

MIB Expression Editor Options

Attribute	Description
MIB Expression	Displays the MIB Expression as it is created. You can place the cursor in the MIB Expression field to specify where you want to add or replace an entry.
MIB Variable	You must select a MIB Variable using the MIB tree. Click the icon to access the MIB tree and navigate to the MIB variable of interest. **Note:** If you do not see a MIB that you recently loaded, wait 1 minute for NNMi to cache the new MIB information, and then open the MIB tree again. After you select a MIB Variable, NNMi displays the MIB Variable's name. If you choose a MIB Variable that has multiple instances, you MUST specify a MIB Filter Variable and MIB Filter. For example, because a node can have multiple interfaces, MIB Variables containing interface information have multiple instances, one for each interface. You are required to provide a MIB Filter value to select the interfaces you want NNMi to poll. If you do not specify a MIB Filter Variable and MIB Filter, NNMi assumes the MIB variable is non-repeating. Click here for more information. For example, if you want to always gather additional HOST-RESOURCES-MIB status information about COM (communication) port devices, you would define the following: • MIB Expression: `hrDeviceStatus` • MIB Filter Variable: `hrDeviceDescr` • MIB Filter: `COM*` See "Create a Policy" on page 449 for more information about the MIB Filter.
Constant Value	A numeric value to be used in the calculation for the MIB Expression. For example, you might want to include 100 as a constant when calculating percentages.
Enter	Includes the Constant Value in the MIB Expression.
+	Adds the results.
-	Subtracts the results.
*	Multiplies the results.
/	Divides the results.
<>	Changes the selected operator (+, -, *, and /) to the operator that appears next in sequence (from left to right) in the MIB Expression Editor. (The example below shows the operator sequence in the MIB Expression Editor.)

MIB Expression Editor Options, continued

Attribute	Description
	For example, if you place your cursor at an add (+) operator in the MIB Expression, the MIB Expression Editor changes the add (+) operator to the minus (-) operator. If you place your cursor at the divide (/) operator in the MIB Expression as shown in the example below, the MIB Expression Editor changes the operator to the add (+) operator.  * Expression: (ifInOctets / 100) MIB Variable Constant Value Enter + - * / <> Delete OK Clear Cancel When using the <> (Change Operator) button, note the following: • You must select an operator in the MIB Expression before using the Change Operator (<>) button. • You can replace a MIB Variable with another MIB Variable or with a Constant. You can replace a Constant value with a MIB Variable or Constant.
Delete	Deletes the entry that is selected. If no entry is selected, NNMi deletes the last entry in the MIB Expression.
OK	Closes the MIB Expression Editor and saves your changes.
Clear	Removes any entries in the MIB Expression.
Cancel	Closes the MIB Expression Editor without saving your changes.

Configure Threshold Information for a Custom Poller Collection

Thresholds specify the high and low values from the MIB Expression results that indicate a High Threshold situation or Low Threshold situation. NNMi administrators can configure NNMi to change the associated High State and Low State for the Custom Polled Instance and generate an incident based on the Custom Polled Instance's State.

> **Tip:** Check to see if the threshold you want is already defined. See "About Threshold Settings Provided by NNMi" on page 354.

When configuring Threshold settings, note the following:

- If a polled value is between the high range and the low range, the Polled Instance state is Normal.
- You can configure Comparison Maps, which also contribute to State calculations. If you configure both Thresholds and Comparison Maps, NNMi first checks the Threshold settings to determine State values. If the threshold evaluates to non-Normal, NNMi uses the Threshold settings to determine State values. if the Threshold evaluates to Normal, NNMi checks for a non-Normal State using any Comparison Map configuration. See "Configure Comparison Maps for a Custom Poller Collection" on page 446 for more information about configuring Comparison Maps.
- The MIB Expression must evaluate to a numeric type. (OCTET STRING type is not supported.)
- When evaluating Threshold configurations with MIB Expressions that include one or more MIB Variables of type Counter or Counter64, NNMi evaluates the MIB Variable value using the difference in value between the most recent poll and the poll before it.

To configure thresholds for a MIB Variable:

1. *Prerequisite:* You must specify the MIB Expression you want to poll. See "Specify the MIB Variable Information for a Custom Poller Collection" on page 429 for more information.
2. Navigate to the Custom Poller Collection form.

   a. From the workspace navigation panel, select the **Configuration** workspace.

   b. Expand the **Monitoring** folder.

   c. Select the **Custom Poller Configuration** form.

   d. Select the **Custom Poller Collections** tab.

   e. Do one of the following:

      - To create a collection, click the New icon.
      - To edit a collection, double-click the row representing the configuration you want to edit.

   a. Locate the **Thresholds** section of the form.
3. Make your configuration choices (see table).
4. Click **Save and Close** to close the **Custom Poller Collection** form.
5. Complete the configuration for this Custom Poller Collection configuration, if you have not already done so:
6. Click **Save and Close** to close the **Custom Poller Configuration** form.

High Threshold Attributes for a Custom Polled Instance

Monitored Attribute	Description
Threshold Setting Type	Select one of the following: • **Count** to configure count-based thresholds. Count-based threshold settings enable you to determine as soon as a threshold is crossed (for example, the results of polling the MIB Expression are above 90

High Threshold Attributes for a Custom Polled Instance, continued

Monitored Attribute	Description
	percent for 4 consecutive polls). See "Examples of Count-Based Threshold Monitoring" on page 364 for more information. • **Time** to configure time-based thresholds. Time-based threshold settings enable you to determine whether a threshold is crossed within a particular duration of time (for example, the results of polling the MIB Expression are above 90 percent for 20 out of 30 minutes). See "Examples of Time-Based Threshold Monitoring" on page 368 for more information.
High State	The Custom Polled Instance's State when the results of polling the MIB Expression exceed the specified High Value for the specified Count or Duration. Possible values are: • Normal • Warning • Minor • Major • Critical
High Value	Designate the value that above which becomes a threshold situation. The appropriate value depends on the MIB Expression definition (see "Specify the MIB Variable Information for a Custom Poller Collection" on page 429). For special situations, the following values can be used: • 0.000000000000001 (or 1E-15 in Scientific Notation) for the smallest value greater than zero. • 99.99999999999999 for the highest value less than one hundred. **Note:** If you use the maximum possible value, the High threshold is disabled because it cannot be *crossed*.
High Value Rearm	The High Value Rearm designates the lower boundary of the High Threshold *range of values*. Designate a numeric value appropriate for the MIB Expression definition. After entering a High threshold situation, when a returned value is below the specified High Value Rearm, the following happens: • For Time-Based Thresholds: ▪ The current polling interval does not contribute toward High Duration. ▪ The criteria for High Duration and High Duration Window determine when the High Threshold situation ends.

High Threshold Attributes for a Custom Polled Instance, continued

Monitored Attribute	Description
	• For Count-Based Thresholds: After entering a High threshold situation, when a returned value is below the specified High Value Rearm, the High Threshold situation ends. **Note:** The High Value Rearm must be less than or equal to the High Value and greater than or equal to the Low Value Rearm (if any).
Threshold Setting Type = Count (setting this to ???? disables the High Threshold):	
High Trigger Count	Designate the number of consecutive polling intervals the returned value must be greater than the specified High Value to meet the High Threshold criteria. The default value is 1.
Threshold Setting Type = Time (setting both of these to zero disables the High Threshold):	
High Duration	Designate the minimum time within which the value must remain in the High range before the threshold state changes to High and (optionally) an incident is generated. The High Duration should be equal to or greater than the associated Polling Policy's *Polling Interval* setting, because that is how often NNMi provides a data point. **Note:** The polling interval should be less than or equal to the High Duration. The High Duration should be a multiple of the polling interval. For example, if the polling interval is 5 minutes, use multiples of 5 (10, 15, or 20).
High Duration Window	Designate the window of time within which the High Duration criteria must be met. The value must be greater than 0 (zero) and can be the same as or greater than the High Duration value. NNMi uses a sliding window, meaning that each time the High Window Duration is reached, NNMi drops the oldest polling interval and adds the most recent.

Low Threshold Attributes for a Custom Polled Instance

Monitored Attribute	Description
Threshold Setting Type	Select one of the following: • **Count** to configure count-based thresholds. Count-based threshold settings enable you to determine as soon as a threshold is crossed (for example, the results of polling the MIB Expression are above 90 percent for 4 consecutive polls). See "Examples of Count-Based Threshold Monitoring" on page 364 for more information.

Low Threshold Attributes for a Custom Polled Instance, continued

Monitored Attribute	Description
	• **Time** to configure time-based thresholds. Time-based threshold settings enable you to determine whether a threshold is crossed within a particular duration of time (for example, the results of polling the MIB Expression are above 90 percent for 20 out of 30 minutes). See "Examples of Time-Based Threshold Monitoring" on page 368 for more information.
Low State	The Custom Polled Instance's State when the results of polling the MIB Expression are below the specified Low Value for the specified Count or Duration. Possible values are: • Normal • Warning • Minor • Major • Critical
Low Value	Specify the value that *below which* indicates entering the Low range.The appropriate value depends on the MIB Expression definition (see "Specify the MIB Variable Information for a Custom Poller Collection" on page 429). The Low Value must be less than or equal any specified High Value. **Note:** If you use the minimum possible value, the threshold is disabled because it cannot be *crossed*.
Low Value Rearm	The Low Value Rearm designates the upper boundary of the Low Threshold *range of values*. Designate a numeric value appropriate for the MIB Expression definition. After entering a Low threshold situation, when a returned value is above the specified Low Value Rearm, the following happens: • For Time-Based Thresholds: ▪ The current polling interval does not contribute toward Low Duration. ▪ The criteria for Low Duration and Low Duration Window determine when the Low Threshold situation ends. • For Count-Based Thresholds: After entering a Low threshold situation, when a returned value is above the specified Low Value Rearm, the Low Threshold situation ends. **Note:** The Low Value Rearm must be greater than or equal to the

Low Threshold Attributes for a Custom Polled Instance, continued

Monitored Attribute	Description
	Low Value and less than or equal to the High Value Rearm (if any).
Threshold Setting Type = Count (setting this to ???? disables the Low Threshold):	
Low Trigger Count	Designate the number of consecutive polling interval the returned value must be less than the specified Low Value to meet the Low Threshold criteria. The default value is 1.
Threshold Setting Type = Time (setting both of these to zero disables the Low Threshold):	
Low Duration	Designate the minimum time within which the value must remain in the Low range before the threshold state changes to Low and (optionally) an incident is generated. **Note:** The polling interval should be less than or equal to the Low Duration. The Low Duration should be a multiple of the polling interval. For example, if the polling interval is 5 minutes, use multiples of 5 (10, 15, or 20).
Low Duration Window	Designate the window of time within which the Low Duration criteria must be met. **Note:** The value must be greater than 0 (zero) or equal to the Low Duration value. NNMi uses a sliding window, meaning that each time the Low Window Duration criteria is met, NNMi drops the oldest polling interval and adds the most recent. See "Examples of Time-Based Threshold Monitoring" on page 368 for more information.

Configure Comparison Maps for a Custom Poller Collection

Prerequisite: You must know the valid values that might be returned when the MIB Expression is polled.

Custom Poller enables you to map the returned value of a MIB Expression to a Custom Polled Instance *State*. These values are used to determine the High State and Low State of the Custom Polled Instance. NNMi administrators can configure NNMi to generate an incident when the Custom Polled Instance's State changes. For example, you might want the `hrDeviceStatus` value of **5** (or lower) to be mapped to a **Critical** State. This means that NNMi changes the State of the Polled Collection Instance to **Critical** each time the `hrDeviceStatus` returns a value of **5** when polled.

When configuring Comparison Maps, note the following:

- NNMi applies the Comparison Maps according to the Ordering number defined. The first comparison criteria met defines the State for the Polled Instance.
- You can configure Thresholds, which also contribute to State calculations. If you configure both Thresholds and Comparison Maps, NNMi first checks the Threshold settings to determine State values. If the threshold evaluates to non-Normal, NNMi uses the Threshold settings to determine State values. if the Threshold evaluates to Normal, NNMi checks for a non-Normal State using any Comparison Map configuration. See "Configure Threshold Information for a Custom Poller Collection" on page 441 for more information about configuring thresholds.
- When evaluating Threshold configurations with MIB Expressions that include one or more MIB Variables of type Counter or Counter64, NNMi evaluates the MIB Variable value using the difference in value between the most recent poll and the poll before it.

To configure Comparison Maps for a MIB Expression:

1. *Prerequisite:* You must specify the MIB Expression you want to poll. See "Specify the MIB Variable Information for a Custom Poller Collection" on page 429 for more information.
2. Navigate to the **Custom Poller Collection** form.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Expand the **Monitoring** folder.
   c. Select the **Custom Poller Configuration** form.
   d. Select the **Custom Poller Collections** tab.
   e. Do one of the following:
      - To create a collection, click the ✱ New icon.
      - To edit a collection, double-click the row representing the configuration you want to edit.
3. Locate the **Comparison Maps** tab.
4. Do one of the following:
   - To create a Comparison Map, click the ✱ New icon.
   - To edit a Comparison Map, double-click the row representing the configuration you want to edit.
5. Make your configuration choices (see table).
6. Click **Save and Close** to close the **Custom Poller Collection** form.
7. Complete the configuration for this Custom Poller Collection configuration, if you have not already done so:
8. Click **Save and Close** to close the **Custom Poller Configuration** form.

> **Note:** Each time you save a Comparison Maps configuration, NNMi suspends Custom Polling for the Custom Poller Collection. When you finish making your Comparison Mapping changes, set the Active State to **Active** for each of the policies in the Custom Poller Collection that you want to be in use. See "Create a Policy" on page 449 for more information.

State Mapping Attributes

Attribute	Description
Ordering	The order in which the State mapping (Comparison Maps) operations should be performed. **Note:** NNMi uses the Ordering value to determine which State mapping to use. The lower the number, the higher the priority. For example, 1 is the highest priority.
Comparison Operator	Operator used to evaluate the Comparison Value and subsequently determine its State. For example, the < (less than) Comparison Operator indicates the polled value must be less than the Comparison Value specified to change the Custom Poller Polled Instance to the specified State value. Possible Comparison Operator values are: • **<** (Less than) • **<=** (Less than or equal to) • **=** (Equal to) • **!=** (Not equal to) • **>** (Greater than) • **>=** (Greater than or equal to) • **is null** (Null or unavailable) • **is not null** (Contains a value) • **default** (Sets the State when no matches are found using the other Comparison Operators) **Note:** Ordering for the **default** Comparison Operator must be the last.
Comparison Value	The value returned when the MIB Expression is evaluated when polled.
State Mapping	The State to assign to the Custom Poller Polled Instance when the polled value is returned. For example, each time the value **3** (warning) is returned when NNMi polls hrDeviceStatus, you can specify that you want NNMi to change the State of the Polled Instance to **Warning**. Possible State values for a *Polled Instance* (Threshold = High State/Low State; or Comparison Map = State Mapping) are: ✓ Normal ▲ Warning ⚠ Minor

State Mapping Attributes, continued

Attribute	Description
	▽ Major ⊗ Critical

Create a Policy

Prerequisite: Make sure that the Node Group has been created to which you want to apply the Custom Polling Policy. See Define Node Groups for more information about creating Node Groups.

You can create one or more policies for a Custom Poller Collection. When configuring a Custom Poller Policy, you define which MIB variable or variables NNMi gathers from members of a specific Node Group.

If you configure more than one Policy per Collection, each Policy must be for a different Node Group.

The Management Mode setting for the node is used to determine whether NNMi collects Custom Poller information for the node regardless of the Management Mode for any associated interfaces. See Management Mode and Custom Poller for example scenarios.

> **Note:** These scenarios assume that the Custom Poller MIB Expression is configured to access MIBs from the Interface table.

Management Mode and Custom Poller

Node Management Mode	Interface Management Mode	Access Node MIBs	Access Interface MIBs
Not Managed or Out of Service	Not Managed or Out of Service	No	No
Not Managed or Out of Service	Managed	No	No
Managed	Not Managed or Out of Service	Yes	Yes

To configure a Custom Poller Policy:

1. Navigate to the Custom Poller Policies form.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Expand the **Monitoring** folder.
   c. Select the **Custom Poller Configuration** view.
   d. Locate the **Policies** tab.
   e. Do one of the following:

- To create a policy, click the New ✱ icon.
- To edit a policy, double-click the row representing the configuration you want to edit.

2. Make your configuration choices (see table).
3. Click **Save and Close** to return to the **Custom Poller Configuration** form.

   To verify that Custom Poller is working as expected, see the report on the Custom Poller tab in **Help** → **System Information**.

Custom Poller Policy Attributes

Attribute	Description
Name	The Name of the Policy configuration. Alpha-numeric and special characters (~ ! @ # $ % ^ & * ( ) _+ -) are permitted. No spaces are permitted. **Note:** The Policy name appears in any incidents generated as a result of the Collection. Specify a name that will help you to indicate the types of nodes that are polled with this policy.
Ordering	The order in which the Policy should be considered for nodes that appear in multiple Node Groups and therefore might have conflicting Policies. For example, Ordering is used in the following scenario: • Two Policies associated with the same Custom Poller Collection specify `ifOperStatus` as the MIB Expression. • One Policy uses the Routers Node Group and the second Policy uses the Switches Node Group. • Each Policy has a different Polling Interval. In the example scenario above, if a device was in both the Routers Node Group and the Switches Node Group, NNMi would poll the device only one time according to the Policy with the lowest Ordering number.
Collection	Click the Lookup icon and select Show Analysis or Open to display more information about the Custom Poller Collection.
Active State	Use the Active State setting to specify which Custom Poller Policies you want to enable or temporarily disable. The Active State for the associated Custom Collect Policy. Possible values are described below: **Active** - Indicates the Custom Poller Policy is in use. **Note:** At the time the Active State attribute is set to **Active**, NNMi applies the Custom Poller Policy to the nodes in the specified Node Group to determine which instances should be polled. **Inactive** - Indicates the Custom Poller Policy is not in use. NNMi removes all Polled

Custom Poller Policy Attributes, continued

Attribute	Description
	Instances associated with the Policy. **Suspended** - Indicates someone on your team changed this Custom Poller Policy's *Active State* to `Suspended`, or the NNMi administrator disabled Custom Poller in the *Global Control* settings of **Configuration** workspace, **Custom Poller Configuration** form. NNMi suspends polling and retains the most recent State value from before the Policy was suspended.
Node Group	The Node Group to which the Custom Poller Policy applies.
MIB Filter	The MIB Filter value to be used as the filter for determining the Polling Instances. When using a MIB Filter, note the following: • The MIB Filter value must match the return type of your filter variable. For example, because `hrDeviceDescr` is of type String, to poll only those MIBs associated with each node that includes the description for a COM (communication) port, **COM*** would be the MIB Filter for the example MIB Filter Variable `hrDeviceDescr`. • If your MIB Expression includes a MIB Variable that has multiple instances, you MUST specify a MIB Filter Variable and MIB Filter. For example, because a node can have multiple interfaces, MIB variables containing interface information have repeating instances and require you to use a MIB Filter to specify which interfaces you want NNMi to poll. • If your MIB Expression contains more than one MIB Variable with multiple instances, the MIB Filter must apply to each of these MIB Variables. • Valid types for MIB Filter Variables include the following: ▪ INTEGER ▪ UNSIGNED INTEGER ▪ GAUGE ▪ OCTET STRING ▪ IpAddress (IPv4 only) Click here for information about valid values for the MIB Filter Expression. Valid values for MIB Filter include the following: • For numeric values only, you can specify a range using a dash (-). For example 1-6. • For string values only, you can use the wildcard character (*) at either the beginning or end of a string value. For example: *vlan, vlan*, and *vlan*. To match all instances, specify *. • For either numeric or sting values, you can use the Not operator (!) at the beginning of the MIB Filter expression. For example: !1-3, !*vlan, and !vlan.

Custom Poller Policy Attributes, continued

Attribute	Description
	When using MIB Filters, note the following: • NNMi uses exact matches for string comparisons. • String comparisons are case insensitive. • NNMi ignores leading and trailing white spaces • You can specify multiple MIB Filter expressions by separating each MIB Filter using a comma (,) • When you enter multiple MIB Filter expressions, NNMi combines them using the OR operator. • To include the dash (-), asterisk (*), or exclamation (!), or comma (,) in your search, use a leading backslash (\) before the special character.
Polling Interval	The interval in which to perform the Custom Poll.

Create a Report Group (NNM iSPI Performance for Metrics)

Report Groups enable you to define which Custom Poller Collections are reported to NNM iSPI Performance for Metrics. Each Report Group you configure represents a tab in the NNM iSPI Performance for Metrics Report Menu.

> **Caution:** If you delete a Report Group, NNM iSPI Performance for Metrics removes all historical reporting data associated with that Report Group. To retain the historical reporting data, change the Active State of the associated Custom Poller policy to **Suspend**. See "Create a Policy" on page 449 for more information.

To configure a Report Group:

1. Navigate to the **Report Group** form.

   a. From the workspace navigation panel, select the 🔧 **Configuration** workspace.

   b. Expand the **Monitoring** folder.

   c. Select the **Custom Poller Configuration** form.

   d. Select the **Report Groups** tab.

   e. Do one of the following:

      - To create a Report Group, click the New ✳ icon, and continue.

      - To edit a Report Group, double-click the row representing the configuration you want to edit, and continue.

      - To delete a Report Group, select a row, and click the ✖ Delete icon.

2. Make your configuration choices (see table).

3. Click **Save and Close** to return to the **Custom Poller Configuration** form.

4. Create a Report Collection to associate one or more Custom Poller Collections with this Report Group. See "Create a Report Collection (NNM iSPI Performance for Metrics)" below for more information.

   To view the Report Collection configuration associated with a selected Report Group, from the **Custom Poller Collections** or **Report Groups** tab, select **Actions >HP NNM iSPI Performance > Show Report Configuration**. NNMi displays the following information:

   > **Note:** NNMi displays the **Show Report Configuration** menu option only if you have an HP Network Node Manager iSPI Performance for Metrics Software license key installed on the NNMi management server.

   - Report Configuration file name
   - Report Group unique identifier (UUID)
   - Name of the metrics collected by this report configuration

Custom Poller Report Group Attributes

Attribute	Description
Name	Enter the name that you want to appear in the tab in the NNM iSPI Performance for Metrics Report Menu for this Report Group. The name can be up to 255 alphanumeric characters. Spaces are permitted. The following special characters (<, >, ", ', &, \, \, #) are not permitted.

Create a Report Collection (NNM iSPI Performance for Metrics)

Report Collections enable you to specify a Custom Poller Collection to be associated with a Report Group as well as the type of data that is being collected. You can create one or more Report Collections for a Report Group.

> **Caution:** If you delete a Report Collection, NNM iSPI Performance for Metrics removes all historical reporting data for that Report Collection.

To configure a Report Collection:

1. Navigate to the **Report Collection** form.

   a. From the workspace navigation panel, select the **Configuration** workspace.

   b. Expand the **Monitoring** folder.

   c. Select the **Custom Poller Configuration** form.

   d. Locate the **Report Groups** tab.

   e. Do one of the following:
      - To create a Report Group, click the New ✱ icon.
      - To edit a Report Group, double-click the row representing the configuration you want to edit.

2. Select the **Report Collections** tab.
3. Do one of the following:
   - To create a Report Collection, click the New ✱ icon.
   - To edit a Report Collection, double-click the row representing the configuration you want to edit.
   - Make your configuration choices (see table).
4. Click **Save and Close** to return to the **Custom Poller Configuration** form.

   To verify that Custom Poller is working as expected, see the report on the Custom Poller tab in **Help → System Information**.

   To view the Report Collection configuration associated with the selected Report Collection, from the **Custom Poller Collections** or **Report Groups** tab, select **Actions → HP NNM iSPI Performance → Show Report Configuration**. NNMi displays the following information:

   > **Note:** NNMi displays the **Show Report Configuration** menu option only if you have an HP Network Node Manager iSPI Performance for Metrics Software license key installed on the NNMi management server.

   - Report Configuration file name
   - Report Group unique identifier (UUID)
   - Name of the metrics collected by this report configuration

   > **Note:** If the Report Collection displays data as a different type than expected, check the **MIB OID Types** table in the **Configuration** workspace. The NNMi administrator can override the MIB OID Type values generated by Custom Poller. See "Override MIB OID Types" on page 1484 for more information.

Custom PollerReport Collection Attributes

Attribute	Description
Custom Poller Collection	Specifies a Custom Poller Collection that should be associated with the Report Group you are configuring. Click the Lookup icon, and do one of the following: • To specify a Custom Poller Collection, select Quick Find . In the Quick Find dialog, select the Custom PollerCollection of interest. • To create a Custom Poller Collection, click the New ✱ icon. • To edit a Custom Poller Collection, select a row, click the Open icon.

Custom PollerReport Collection Attributes, continued

Attribute	Description
	When specifying a Custom PollerCollection, note the following: ● A Custom Poller Collection can be associated with only one Report Group. ● If you associate more than one Custom Poller Collection with the same Report Group, make sure the combination of Collections will generate a meaningful report. Use the following general guidelines: ■ Select Collections with MIB Variables that are indexed in the same MIB table. For example, you might group a collection that includes power supply information, such as UPS line voltage (upsInputVoltage and upsOuputVoltage), UPS line current (upsInputCurrent and upsOutputCurrent) and UPS line power (upsInputPower and upsOutputPower). ■ Select Collections with MIB Variables that are stored in different MIBs, but that would be useful to visualize together at an aggregate level. For example, you might choose to group power supply line load (upsOutputPercentLoad) and power supply battery temperature (upsBatteryTemperature). ■ Select Collections representing the same index value or similar data across Custom Poller Collections. For example, you might want to examine environment sensor information (such as temperature, humidity, dew point, airflow, and audible sounds such as alarms), in the same report even though this information comes from different MIBs. ● As soon as the Report Collection is saved, NNMi updates the information in the NNM iSPI Performance for Metrics Report Menu.
Report Data Type	Determines how NNM iSPI Performance for Metrics interprets the metrics to be displayed. Possible values include: ● **Gauge** – Represents single non-cumulative values. Examples of Gauge data types include Response Time, Bit Rate, and Temperature. When Gauge data types are aggregated, NNM iSPI Performance for Metrics calculates the minimum, maximum, and average values. ● **Percent** – Represents single non-cumulative values that are formatted with a percent sign (%) and two decimal places. Examples of Percent data types include Utilization and Discard Rate. When Percent data types are aggregated, NNM iSPI Performance for Metrics calculates the minimum, maximum, and average values.. ● **Counter** – Represents incremental values. Examples of Counter data types include byte counts, packet counts, and flow counts. When Counter data types are aggregated, NNM iSPI Performance for Metrics calculates the sum.

Chapter 11

Stop or Start Managing an Object

NNMi administrators can specify that a node, interface, card, address, or node component should no longer be discovered or monitored (Management Mode = Unmanaged) or is out of service (Management Mode = Out of Service). For additional information see the form for each object:

Reasons you might want to change the management mode include:

- The node is temporarily out of service.
- You determine that NNMi should never monitor a particular node, interface, card, IP address, or node component.

NNMi provides two management modes for each object (as described in the table). For more information, see the following topics:

- "View the Management Mode for Objects in Your Network" on the next page
- "How NNMi Assigns the Management Mode to an Interface, Card, Node Component, or Address" on page 461
- "How the NNMi Administrators Change a Management Mode" on page 463
- "Understand the Effects of Setting the Management Mode to Not Managed or Out of Service " on page 464

Management Modes

Name	Description
Node Management Mode or Management Mode	For Node objects, this value is set by the NNMi administrator. The node-level Management Mode affects the Management Mode of objects associated with the node. For interface, card, node component, or address, the Management Mode cannot be set by the NNMI administrator. NNMi calculates value. The Management Mode for an interface, card, or node component is computed based on the Management Mode for the node. The Management Mode value for an address is calculated based on the Management Mode for the associated interface (if any) or based on the Management Mode for the node. Possible values include: **Managed** - Used to indicate a node, interface, or address should be discovered and monitored by NNMi. **Not Managed** - Used to indicate that NNMi should not discover or monitor the object. For example, the object might not be accessible because it is in a private network.

Management Modes, continued

Name	Description
	Out of Service - Used to indicate a node, interface, or address is unavailable because it is out of service. NNMi does not discover or monitor these objects. **Tip**: Some objects have child objects (for example, Nodes contain interfaces, and interfaces can contain IP addresses). To change the Management Mode back to **Managed** or **Inherited** for the selected object and all associated child objects, use the **Actions** → **Management Mode** → **Managed (Reset All)**. **Tip:** You can right-click any object in a table or map view to access the **Actions** menu.
Direct Management Mode	For interfaces, cards, node components, and addresses, this value is set by the NNMi administrator. NNMi uses this value to compute the Management Mode values in the previous row in this table. Possible values include: **Inherited** - For interfaces, cards, and node components, this value is used to indicate that the object should inherit the Management Mode from the node in which it resides. For addresses, this value is used to indicate that the Management Mode should be inherited from the associated interface, if one exists. Otherwise the Management Mode is inherited from the node in which it resides. **Not Managed** - Used to indicate that NNMi should not discover or monitor the object. For example, the object might not be accessible because it is in a private network. **Out of Service** - Used to indicate the object is unavailable because it is out of service. Reasons might include the interface, card, or node component is being repaired or there is a known problem with the address. NNMi does not discover or monitor these objects.

View the Management Mode for Objects in Your Network

NNMi provides the Management Mode workspace so that you can quickly view lists of all nodes, interfaces, cards, addresses, or node components that NNMi is not currently discovering or monitoring. For information about these views:

The following tables describes each possible Management Mode and Direct Management Mode value. The available Management Mode values depend on the object type (node, interface, card, address, or node component). Management Modes are either set automatically by NNMi or set by the NNMi administrators.

Management Mode Values

Object	Value	Description
Node	Managed	Used to indicate that the node should be managed by NNMi. This means it will be discovered and monitored.
Node	Not Managed	Used to indicate you do not plan to manage the node. For example, the node might not be accessible because it is in a private network. NNMi does not discover or monitor these objects.
Node	Out of Service	Used to indicate the node is unavailable because it is out of service. Reasons might include that the device is being repaired or there is a known problem with the device. NNMi does not discover or monitor these objects.

Direct Management Mode Values

Object	Value	Description
Interface, Card, Address, or Node Component	Not Managed	Used to indicate you do not plan to manage the interface, card, address, or node component. After the Direct Management Mode is set to **Not Managed**, NNMi no longer discovers or monitors the object.
Interface, Card, Address, or Node Component	Out of Service	Used to indicate that the object is out of service. NNMi does not discover or monitor these objects. An interface, card, or node component will not be managed again until the Direct Management Mode is set to **Inherited** and its associated node is set to **Managed**. An address will not be managed again until the Management Mode of any associated interface is set to Inherited and the node's Management Mode is set to Managed.
Interface, Card, Address, or Node Component	Inherited	Used to indicate that the object should assume the Management Mode of the node in which it is hosted. **Note**: To manage the interface, card or node component, the Management Mode of the node in which the interface is hosted must be **Managed**.
Address	Inherited	The address assumes the Management Mode of the interface, if any, with which the address is associated. If the address is not associated with an interface, it assumes the Management Mode of the node in which it is hosted. **Note**: To manage the address, the Management Mode of the address' interface, if any, must be calculated to be **Managed**. The Management Mode of the node in which the interface and address are hosted must be set to **Managed**.

Unmanaged Nodes View

The Unmanaged Nodes view identifies all of the nodes with a management mode of either **Not Managed** or **Out of Service**. These are the nodes that are no longer being discovered or monitored.

Use this view to select nodes and change the Management Mode to **Managed**. For information:

For each node, you can identify its overall status (for example, **Normal**, **Warning**, **Minor**, **Major**, **Critical** and **Unknown**), device category (for example, switch), name, system name, management address, system location (the current value of the sysLocation MIB variable), device profile, whether the SNMP Agent is enabled, the date and time the status was last modified, and any notes included for the node.

See "Using the Nodes View" for more information about uses for nodes views.

Related Topics

"How the NNMi Administrators Change a Management Mode" on page 463

"How NNMi Assigns the Management Mode to an Interface, Card, Node Component, or Address" on page 461

"Understand the Effects of Setting the Management Mode to Not Managed or Out of Service " on page 464

"View the Management Mode for Objects in Your Network" on page 457

Unmanaged Interfaces View

Tip: See Interface Form for more information about the attributes that appear in each column in this view.

The Unmanaged Interfaces view identifies all of the interfaces with a Management Mode of **Not Managed** or **Out of Service**. These are the interfaces that are no longer being discovered or monitored.

Use this view to select interfaces and change the Management Mode to **Managed**. For information:

For each interface, you can identify the interface's overall status (for example, **Normal**, **Warning**, **Minor, Major, Critical**, and **Unknown**), administrative state, operational state, the management mode of the interface, the management mode of the associated node, the node on which the interface resides (Hosted on Node), the interface name, type, speed, and alias, the date the interface status and state was last changed, and any notes included for the interface.

See Interfaces View (Inventory) for more information about uses for the interfaces views.

Related Topics

"How the NNMi Administrators Change a Management Mode" on page 463

"How NNMi Assigns the Management Mode to an Interface, Card, Node Component, or Address" on page 461

"Understand the Effects of Setting the Management Mode to Not Managed or Out of Service " on page 464

"View the Management Mode for Objects in Your Network" on page 457

Unmananaged Addresses View

Tip: See IP Address Form for more information about the attributes that appear in each column in this view.

The Unmanaged Addresses view identifies all of the addresses with a Management Mode of **Not Managed** or **Out of Service**. These are the addresses that are no longer being discovered or monitored.

Use this view to select addresses and change the Management Mode to **Managed**. For information:

For each IP address, you can identify its status, state, management mode, the management mode of its associated node, the IP address value, the name of the interface on which the address resides (**In Interface**), the name of the node on which the address resides (**Hosted on Node**), subnet (**In Subnet**) and prefix length (**PL**), the date and time in which the status was last modified, and any notes included for the IP address.

See IP Addresses View (Inventory) for more information about uses for address views.

Related Topics

"How the NNMi Administrators Change a Management Mode" on page 463

"How NNMi Assigns the Management Mode to an Interface, Card, Node Component, or Address" on the next page

"Understand the Effects of Setting the Management Mode to Not Managed or Out of Service " on page 464

"View the Management Mode for Objects in Your Network" on page 457

Unmanaged Cards View

Tip: See Card Form for more information about the attributes that appear in each column in this view.

The Unmanaged Cards view identifies all of the cards with a Management Mode of **Not Managed**or **Out of Service**. These are the cards that are no longer being discovered or monitored.

Use this view to select cards and change the Management Mode to **Managed**. For information:

For each card, you can identify its status, management mode, the management mode of the node on which it resides, the administrative state, the operational state, the name of the node on which the card resides (**Hosted On Node**), the date and time the status was last modified, its name, model, type, serial number, firmware version, hardware version, software version, index, the name of the card on which the card resides, if any, any Redundant Group to which the card belongs, the date and time the state was last modified, the card Description, and any notes included for the card.

Related Topics

"How the NNMi Administrators Change a Management Mode" on page 463

"How NNMi Assigns the Management Mode to an Interface, Card, Node Component, or Address" on the next page

"Understand the Effects of Setting the Management Mode to Not Managed or Out of Service " on page 464

"View the Management Mode for Objects in Your Network" on page 457

Unmanaged Node Components View

Tip: See Node Component Form for more information about the attributes that appear in each column in this view.

The Unmanaged Node Components view identifies all of the Node Components with a Management Mode of **Not Managed** or **Out of Service**. These are the Node Components that are no longer being discovered or monitored.

Use this view to select Node Components and change the Management Mode to **Managed**. For information:

For each Node Component, you can identify its Status, Management Mode, the Management Mode of the node on which it resides, its Name, type, the name of the node on which it resides (**Hosted On Node**), and the date and time the Status was last modified.

Related Topics

"How the NNMi Administrators Change a Management Mode" on page 463

"How NNMi Assigns the Management Mode to an Interface, Card, Node Component, or Address" below

"Understand the Effects of Setting the Management Mode to Not Managed or Out of Service " on page 464

"View the Management Mode for Objects in Your Network" on page 457

How NNMi Assigns the Management Mode to an Interface, Card, Node Component, or Address

NNMi administrators can instruct NNMi to no longer manage an interface, card, node component, or address by setting the *Direct Management Mode* value (see "How the NNMi Administrators Change a Management Mode" on page 463). NNMi then calculates the overall Management Mode based on the current Management Mode of all the associated objects.

For example, if you are specifying the Direct Management Mode for an address, NNMi uses the following values to determine the Management Mode value for the address:

- Direct Management Mode you enter for the address
- Management Mode of the associated interface, if any
- Management Mode of the node that contains the address

The following table lists possible value combinations for each object's management mode.

To check the current Management Mode setting for objects see "View the Management Mode for Objects in Your Network" on page 457 and .

Interface, Card, and Node Component

Management Mode (Node) Calaculated by NNMi	Direct Management Mode (Interface, Card, or Node Component)	Management Mode (Interface, Card, or Node Component)
Managed	Inherited	Managed
Not Managed	Inherited	Not Managed
Out of Service	Inherited	Out of Service
Managed	Not Managed	Not Managed
Not Managed	Not Managed	Not Managed
Out of Service	Not Managed	Not Managed
Managed	Out of Service	Out of Service
Not Managed	Out of Service	Out of Service
Out of Service	Out of Service	Out of Service

Address

Management Mode (Node)	Direct Management Mode (Interface)	Direct Management Mode (Address)	Management Mode (Address)
Managed	Inherited	Inherited	Managed
Not Managed	Inherited	Inherited	Not Managed
Out of Service	Inherited	Inherited	Out of Service
Managed	Not applicable*	Inherited	Managed
Not Managed	Not applicable*	Inherited	Not Managed
Out of Service	Not applicable*	Inherited	Out of Service
Managed	Not Managed	Inherited	Not Managed
Not Managed	Not Managed	Inherited	Not Managed
Out of Service	Not Managed	Inherited	Not Managed
Managed	Not Managed	Not Managed	Not Managed
Not Managed	Not Managed	Not Managed	Not Managed
Out of Service	Not Managed	Not Managed	Not Managed

Address, continued

Management Mode (Node)	Direct Management Mode (Interface)	Direct Management Mode (Address)	Management Mode (Address)
Managed	Not applicable*	Not Managed	Not Managed
Not Managed	Not applicable*	Not Managed	Not Managed
Out of Service	Not applicable*	Not Managed	Not Managed
Managed	Out of Service	Inherited	Out of Service
Not Managed	Out of Service	Inherited	Out of Service
Out of Service	Out of Service	Inherited	Out of Service
Managed	Out of Service	Out of Service	Out of Service
Not Managed	Out of Service	Out of Service	Out of Service
Out of Service	Out of Service	Out of Service	Out of Service
Managed	Not applicable*	Out of Service	Out of Service
Not Managed	Not applicable*	Out of Service	Out of Service
Managed	Not applicable*	Out of Service	Out of Service

* Used to indicate there is no associated interface

How the NNMi Administrators Change a Management Mode

Caution: (*NNMi Advanced - Global Network Management feature*) If your NNMi console is a Global Manager and the selected object is being managed by a Regional Manager (another NNMi management server in your network environment), you cannot change the Management Mode setting unless you log on to the Regional Manager (NNMi management server).

The NNMi administrator can change the Management Mode of a node, interface, card, node component, or IP address in one of the following ways:

- Open the object's form, do one of the following, and then select **File** → **Save and Close**:
  - Use the Management Mode attribute's drop-down menu to choose an available Management Mode for that object.
  - Use **Actions** → **Management Mode** and choose an available Management Mode for that object.

    Tip: You can right-click any object in a table or map view to access the **Actions** menu.

Note: If you are updating the Direct Management Mode for an interface, card, node component, or address, NNMi also updates its Management Mode value after you reopen or refresh the form.

- Open a view that contains the object and do the following:

  a. Select the object of interest:

     - In a table view, select the row representing the object information.
     - In a map view, single-click the object.

  b. Select **Actions** → **Management Mode** and choose an available Management Mode for that object.

     Tip: Some objects have child objects (for example, Nodes contain interfaces, and interfaces can contain IP addresses). To change the Management Mode back to **Managed** or **Inherited** for the selected object and all associated child objects, use the **Actions** → **Management Mode** → **Managed (Reset All)**.

Note: The NNMi administrator can also change the management mode of a node or interface using the nnmmanagementmode.ovpl command.

Make sure you review this information: "Understand the Effects of Setting the Management Mode to Not Managed or Out of Service " below.

Related Topics:

"How NNMi Assigns the Management Mode to an Interface, Card, Node Component, or Address" on page 461

"View the Management Mode for Objects in Your Network" on page 457

Understand the Effects of Setting the Management Mode to Not Managed or Out of Service

NNMi administrators can instruct NNMi to no longer manage an interface, card, node component, or address by selecting a *Management Mode* value on the object's form or by using **Actions** → **Management Mode**. See "How the NNMi Administrators Change a Management Mode" on the previous page.

> **Tip:** You can right-click any object in a table or map view to access the **Actions** menu.

The results of setting the management mode to **Not Managed** or **Out of Service** for an object, depends on whether you are setting the value for a node, interface, address, card, or node component:

- **Nodes: Management Mode**

  For nodes, setting the Management Mode to **Not Managed** or **Out of Service** has the following effects:

  - No incidents are generated for the node
  - The node's SNMP Agent is excluded from fault polling.
  - The node's interfaces or addresses are excluded from fault and performance polling.
  - NNMi quits gathering Node Component data.
  - NNMi deletes all Polled Instances associated with the **Not Managed** or **Out of Service** node.
  - The Active State for any Custom Poller Nodes associated with the **Not Managed** or **Out of Service** node becomes **Inactive**.
  - The node is removed from any associated Router Redundancy Groups.
  - Traps related to the node, interface, card, node component, or address, (for example, coldStart or warmStart) are not stored.
  - The node is excluded from discovery.
  - **Actions** → **Polling** → **Configuration Poll** is no longer available for this node.
  - The status of a node is set to **No Status**.
  - **Actions** → **Polling** → **Status Poll** is no longer available for the node or incident related to that node.

- **Interfaces: Direct Management Mode**

  For interfaces, setting the Direct Management Mode to **Not Managed** or **Out of Service** has the following effects:

  - No incidents are generated for the interface.
  - The interface and any related addresses are excluded from fault and performance polling.
  - The Administrative State and Operational State of the interface are set to **Not Polled.**
  - The Status of the interface is set to **No Status**.
  - Traps related to the interface (for example, LinkUp or LinkDown), will not be stored.

- **IPv4 / IPv6 Addresses: Direct Management Mode**

  For addresses, setting the Direct Management Mode to **Not Managed** or **Out of Service** has the following effects:

  - No incidents are generated for the address.
  - The State of the address is set to **Not Polled**.
  - The address is excluded from fault and performance polling.
  - Traps related to the address are not stored.

- **Cards and Node Components: Direct Management Mode**

  Cards and Node Components, setting the Direct Management Mode to **Not Managed** or **Out of Service** has the following effects:

  - No incidents are generated for the card or node component.
  - The State of the object is set to **Not Polled**.

- The card or node component is excluded from fault and performance polling.
- The Status remains set to the last known Status value.
- Traps related to the card or node component are not stored.

NNMi provides the Management Mode workspace so that you can quickly view lists of all nodes, interfaces, cards, addresses, or node components that NNMi is not currently discovering or monitoring. For information about these views:

To change the Management Mode back to **Managed**, use the **Actions** → **Management Mode** → **Managed**.

Tip: Some objects have child objects (for example, Nodes contain interfaces, and interfaces can contain IP addresses). To change the Management Mode back to **Managed** or **Inherited** for the selected object and all associated child objects, use the **Actions** → **Management Mode** → **Managed (Reset All)**.

Chapter 12

Configuring the NNMi User Interface

NNMi enables an NNMi administrator to configure the following global user interface features:

- The console timeout interval
- The initial map view to display in the Topology Maps workspace
- Whether NNMi displays unlicensed features that require a special license, such as NNMi Advanced

For information about the additional user interface configurations available, including configuring Node Group map settings, setting the default values for maps and Line Graphs, and configuring menus and menu items:

> **Note:** If you are using multiple tentants, you might want to remove the Nodes Group view from the NNMi console. See the "NNMi Console" chapter of the *HP Network Node Manager i Software Deployment Reference* for more information.

To configure user interface features, do the following:

1. Navigate to the **User Interface Configuration** form.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Click to expand **User Interface**.
   c. Select **User Interface Configuration**.
2. Make your Global Control configuration choices (see the Global Control Attributes table).
3. Make your additional configuration choices. Click here for a list of choices .
4. Click **Save and Close** to apply your changes.
5. To apply your Console Timeout or Initial View configuration changes, sign out of the NNMi console. After restarting the console, your changes should take effect.

Global Control Attributes for User Interface Configuration

Attribute	Description
Console Timeout	NNMi's default session inactivity timeout value is 18 hours. Use this attribute to change the timeout interval in days, hours, and minutes. **Note:** The minimum timeout value is 1 minute. After this period, if no mouse movement occurs, the consoles locks and the user is prompted to sign in again.

Global Control Attributes for User Interface Configuration, continued

Attribute	Description
	Tip: If your network operation center (NOC) has a large screen where a map of the most important nodes is continuously displayed, use a launched view. See "Launch a Troubleshooting Workspace View" on page 1519.The map automatically updates every 30 seconds. (If you are using Mozilla Firefox, also see Configure Mozilla Firefox Timeout Interval.)
Initial View	Use this attribute to specify the initial view to be automatically displayed in the NNMi console by default. When selecting a view from the drop-down menu list, note the following: • Use the value **None (blank)** to specify that you do not want a default view automatically displayed by default. • If the Node Group you select has been removed, NNMi uses None (blank view). • To select a Node Group map you have created: ▪ *Prerequisite*. Use the **Node Group Map Settings** configuration workspace to create a Node Group map and enter a Topology Ordering number that lists the Node Group map as the first or last map in the Topology Maps workspace. See "Configure Basic Settings for a Node Group Map" on page 489 for more information. ▪ For the **Initial View** attribute: ○ If you placed the Node Group map as the first entry in the Topology Maps workspace, select **First Node Group in Topology Maps workspace.** ○ If you placed the Node Group map as the last entry in the Topology Maps workspace, select **Last Node Group in Topology Maps workspace**.
Default Author	The Default Author attribute specifies the Author attribute NNMi should use by default when you create a new instance of an object in NNMi. For example you might create a new incident configuration. The Author attribute identifies who provided that instance of an object. The Author attribute value is also useful for filtering objects in certain views and when using the NNMi Export/Import feature. Either keep the Default Author value of **Customer** or enter an Author attribute value representing you or your organization. The Default Author value you specify then appears in the Author selection list in any appropriate form and appears by default as the Author value when you create a new instance of an object. See Author form for important information.

Global Control Attributes for User Interface Configuration, continued

Attribute	Description
Enable URL Redirect	Before enabling URL Redirect, verify that the NNMi management server's official Fully Qualified Domain Name (FQDN) is set correctly and the DNS name is resolvable from any remote systems that need to access the NNMi management server. If the official FQDN does not meet these requirements, users will view errors when trying to access the NNMi console. To view the NNMi management server's official FQDN, do one of the following: • Select **Help** → **System Information** and click the **Server** tab. • Use the nnmhealth.ovpl command line tool. • Use the nnmofficialfqdn.ovpl command line tool. **Tip:** To change the official FQDN, use the nnmsetofficialfqdn.ovpl command line tool. When ☑ URL Redirect is enabled, a user can sign into the NNMi console using any hostname (*not case-sensitive*) or IP address that is valid for the NNMi management server. (*NNMi Advanced's Global Network Management feature or HP Network Node Manager i Software Smart Plug-ins (iSPIs)*) For environments configured with Single Sign-On (SSO) among multiple servers (which normally requires users to provide the official Fully Qualified Domain Name (FQDN) that was configured during NNMi installation), this attribute enables NNMi to redirect URLs that contain the IP address or any hostname associated with the NNMi management server to the official FQDN. For more information, see the “Using Single Sign-On with NNMi” chapter in the *HP Network Node Manager i Software Deployment Reference* (available at: `http://h20230.www2.hp.com/selfsolve/manuals`). **Note:** All NNMi management servers participating in Global Network Management or Single Sign-On (SSO) must have synchronized time stamps.
Show Unlicensed Features	By default, NNMi displays menus, views, and workspaces that require an additional license. If you do not have the required license, NNMi labels these features as `Unlicensed` or `Evaluation`. `Evaluation` indicates the License Type is `Instant-On` or `Temporary`. To determine which Unlicensed or Evaluation features could be displayed in your NNMi console, click here for more information. • Access **Help** → **Documentation Library** → **Release Notes**and click the **Licensing** link. • Access **Help** → **System Information** and click the **Extension** tab. • Access **Help** → **System Information** and click the **Product** tab and click the **View Licensing Information** button.

Global Control Attributes for User Interface Configuration, continued

Attribute	Description
	See "Purchase an HP Network Node Manager i Smart Plug-in" on page 1486 for more information about possible HP Smart Plug-ins. To hide Unlicensed or Evaluation features from the NNMi console, clear the **Show Unlicensed Features** □ check box. (Recommended if you do not plan to install a permanent license for these features.) To display Unlicensed or Evaluation features in the NNMi console, select the **Show Unlicensed Features** ☑ check box.
Enable Table Row Shading	If enabled ☑, NNMi color-codes each row of an incident view according to the incident status. See About Status Color for more information about status color. If disabled □, incident views are not color-coded.

Registration Attributes for User Interface Configuration

Attribute	Description
Last Modified	Indicates the last date and time that any of the user interface attributes were modified.

NNMi also enables you to configure features specific to Node Group Maps. See "Define Node Group Map Settings" on page 488 for more information.

Define Default Map Settings

Default Map Settings define settings for all of your Node Group Maps.

> **Note:** You can override Default Map Setting using the **Node Group Map Settings** Configuration workspace. See "Configure Basic Settings for a Node Group Map" on page 489 for more information.

To configure Default Map Settings, do the following:

1. Navigate to the **User Interface 'Configuration**form.
   a. From the workspace navigation panel, select the **Configuration** workspace:
   b. Click to expand **User Interface**.
   c. Select **User Interface Configuration**.
2. Navigate to the **Default Map Settings** tab.
3. Make your configuration choices (see the Default Map Settings Attributes table).
4. Click **Save and Close** to return to the **User Interface Configuration** form.
5. Click **Save and Close** to save and apply your changes.

Default Map Settings Attributes

Attribute	Description
Map Refresh Interval	Specifies the refresh interval for Status Refresh.
Maximum Number of Displayed Nodes	Use this attribute to change the maximum number of nodes to be displayed on a map. Note the following: • If you change the default value to display a large number of nodes at one time, you might need to re-adjust this number if maps are taking longer than expected to display. • In Layer 2 and Layer 3 Neighbor views, NNMi adds nodes one hop at a time. If NNMi finds a large number of nodes in a single hop, the number of nodes might exceed the maximum number specified. • The **Initial Discovery Progress** map provided by NNMi displays a maximum number of 100 nodes. The Maximum Number of Displayed Nodes that you specify does not change the maximum number of nodes for this map. • The **Network Overview** map provided by NNMi displays a maximum of 250 nodes by default. The Maximum Number of Displayed Nodes that you specify does not change the maximum number of nodes for this map. However, the NNMi administrator can change the maximum number of nodes displayed using a configuration file. See the "NNMi Console" chapter in the *HP Network Node Manager i Software Deployment Reference* for more information. **Note:** This number applies to the total number of nodes within the Node Group, including the nodes in any Child Node Groups displayed on the map.
Maximum Number of Displayed End Points	Use this attribute to change the maximum number of end points to be displayed on a map. **Note:** If you change the default value to display a large number of end points at one time, you might need to re-adjust this number if maps are taking longer than expected to display.
Multiconnection Threshold	Use this attribute to change the number of connections that must exist between two objects before NNMi displays the connections as one thick line on a map (known as a multiconnection). When this number of connections is reached, NNMi displays the connections as one thick line on all maps except Path View maps. **Note:** To display the Interface objects and each connection, double-click

Default Map Settings Attributes , continued

Attribute	Description
	the line representing the multiconnection.
Indicate Key Incidents	In the Node Group map, NNMi can enlarge the map symbol of any node associated with a **Key Incident**[1]. Uers can click the **Indicate Key Incidents** button in the map view toolbar to toggle this feature on and off (see Using the View Toolbars: Node Group Map Toolbar Icons): 🔑 (on) = When the this Node Group map opens, NNMi enlarges any objects on a Node Group map that are Source Objects for a **Key Incident**[2]. (For example, when viewing the Node Group map, NNMi enlarges any node on a Node Group map that has an open root cause incident associated with it.) 🔑 (off) = When the this Node Group map opens, NNMi does not indicate the objects on a Node Group map that are Source Objects for a **Key Incident**[3]. NNMi administrators can override this default setting for a particular Node Group map, when you "Configure Basic Settings for a Node Group Map" on page 489. See Node Group Maps and Key Incident Views for more information.

Configure Default Settings for Line Graph

NNMi enables you to configure default settings for Line Graphs displayed through the Actions menu.

Note: NNMi provides a set of Line Graphs for node and interface objects that are accessible from the Actions menu. As an NNMi administrator you can configure additional Line Graphs using the **Menu Items**option of the **User Interface** workspace. See "Configure SNMP Line Graph Actions" on page 1437 for more information.

To configure default settings for Line Graphs:

1. Navigate to the **Default Line Graph Settings** tab of the **User Interface Configuration** form.
   a. Navigate to the **Configuration** workspace.

[1] Incidents with both: (1) Severity = other than Normal. (2) Correlation Nature = equal to Root Cause, Service Impact, Stream Correlation, Rate Stream Correlation, Info, or None.
[2] Incidents with both: (1) Severity = other than Normal. (2) Correlation Nature = equal to Root Cause, Service Impact, Stream Correlation, Rate Stream Correlation, Info, or None.
[3] Incidents with both: (1) Severity = other than Normal. (2) Correlation Nature = equal to Root Cause, Service Impact, Stream Correlation, Rate Stream Correlation, Info, or None.

b. Click to expand **User Interface**.

c. Select **User Interface Configuration**.

d. Navigate to the **Default Line Graph Settings** tab.

2. Provide the default settings for all Line Graphs (see the Default Line Graph Settings table).

3. Click **Save and Close** to the **User Interface Configuration** form.

4. Click **Save and Close** to save and apply your changes.

Default Line Graph Settings

Attribute	Description
Default Number of Lines	The Default Number of Lines determines the initial number of lines that are displayed on each Line Graph. **Note**: If more lines than this initial number are available, the user can choose to display additional lines while viewing the graph. You can override this number for an individual graph. See "Configure SNMP Line Graph Actions" on page 1437 for more information.
Default Maximum Time Range (Hours)	The maximum time period in hours in which to retain the Line Graph data point sets. When the Maximum Time Range number is reached, NNMi discards the oldest data point sets so that it can display the most recent data for the time range you specify. For example, if you enter 24 hours, when 24 hours has passed, NNMi removes data starting with the initial data point set so that it can display data for the most recent 24-hour interval. Enter a decimal number indicating the maximum number of hours in which to retain the data. If you do not specify a Maximum Time Range or if you specify 0 (zero), NNMi determines the best setting for the Maximum Time Range based on the Polling Interval specified. If you do not specify a Default Maximum Time Range or set the Default Maximum Time Range to 0 (zero), and you do not specify a Default Polling Interval, NNMi determines the best settings for each so the data fits into the Line Graph displayed. You can override this number for an individual graph. See "Configure SNMP Line Graph Actions" on page 1437 for more information.
Default Update Interval (Seconds)	The Default Update Interval determines how often the NNMi management server polls for the most recent set of data points to be displayed in a Line Graph. **Note**: This Default Polling Interval does not affect the polling intervals set for the NNMi State Poller. Enter the number of seconds in which NNMi should poll for graph data. If you do not specify an Polling Interval, NNMi determines the best setting for the Polling Interval based on the Maximum Time Range specified. If you do not specify an Polling Interval and you do not specify a Maximum Time

Default Line Graph Settings, continued

Attribute	Description
Default Number of Lines	The Default Number of Lines determines the initial number of lines that are displayed on each Line Graph. **Note**: If more lines than this initial number are available, the user can choose to display additional lines while viewing the graph. You can override this number for an individual graph. See "Configure SNMP Line Graph Actions" on page 1437 for more information.
	Range or if you set the Maximum Time Range to 0 (zero), NNMi determines the best setting for each so the data fits into the Line Graph displayed. When viewing a Line Graph, the user can temporarily change the Polling Interval in a Line Graph. After a graph is re-opened, the Polling Interval returns to this default value. At each Polling Interval, the NNMi management server performs an ad-hoc SNMP query to obtain the most current data.

Customize Device Profile Icons

NNMi enables you to customize the icons associated with a Device Profile or specific Nodes. These icons appear in table views, menu items, and as foreground images on an NNMi topology map.

Note: NNMi provides a "missing" icon (🖼) to indicate a Device Profile icon is not available. Reasons NNMi displays a "missing" icon includes 1) the icon has been deleted or 2) the icon does not exist.

- "Add Device Profile Icons" below
- "View the Device Profile Icons Available" on page 477
- "Change the Image for a Specified Icon" on page 478
- "Configure Device Family Icons" on page 480
- "Configure Device Vendor Icons" on page 481
- "Configure Device Category Icons" on page 482
- "Configure the Device Profile Icon for Specified Nodes" on page 479

Add Device Profile Icons

NNMi enables the NNMi administrator to customize the icons associated with a Device Profile or specific Nodes. These icons appear in table views, menu items, and as foreground images on an NNMi topology map.

You can specify icons for a specific Node, Device Family, Device Vendor, or Device Category. NNMi determines the icon to use in the following order of precedence:

- Specified Node Icons
- Device Family Icons
- Device Vendor Icons
- Device Category Icons

> **Tip:** You can use a command line tool to list, create, update, and delete the icons that you load into the NNMi database. See nnmicons.ovpl.

If you delete an icon from the NNMi database, the icon *Name* remains associated with any Device Profile's Device Family, Device Category, or Device Vendor attribute to which that icon was previously assigned. NNMi displays the `missing_image` icon for the affected items until the NNMi administrator updates the specification to an existing icon.

To add icons to the NNMi database:

1. Navigate to the **Icons** option under the **User Interface** folder
   a. Navigate to the **Configuration** workspace.
   b. Click to expand **User Interface**.
   c. Select **Icons**.
2. Provide settings for the icons listed. (see the Device Profile Icons table).
3. Click **Save and Close** to the **Icon** form.
4. Click **Save and Close** to save and apply your changes.
5. If you change your mind, see "Change the Image for a Specified Icon" on page 478.

Device Profile Icons

Attribute	Description
Name	Enter a unique name that identifies the icon. Type a maximum of 64 characters. Alpha-numeric and special characters (~ ! @ # $ % ^ &amp; * ( ) _+ -) are permitted. Spaces are not permitted. **Note**: To enable you to filter the Icons table view by vendor, include the vendor name and the Device Profile attribute for which the icon will be used. For example, for a Device Family icon, you might use the following format: • *<unique_information>*-*<vendor_name>*-`family` For example, for a Device Vendor icon, you might use the following format: • *<unique_information>*-*<vendor_name>*-`vendor` For example, for a Device Category icon, you might use the following format: • *<unique_information>*-*<vendor_name>*- -`category` **Tip**: See the **Name** drop-down menu for example names of icons provided by HP.
Description	Provide additional information that you want to store about the icon. Type a maximum of 2048 characters. Alpha-numeric, spaces, colons (:), and special characters (~ ! @ # $ % ^ & * ( ) _+) are permitted.
Author	See Author form for important information. **Caution**: It is recommended that you create new icon objects rather than modify icons provided by HP. If the Author attribute value is **HP Network Node Manager**, any changes are at risk of being overwritten in the future. Also see: "Export/Import Behavior and Dependencies" on page 1579 Click the Lookup icon and select Show Analysis to display details about the currently selected Author, select Quick Find to access the list of existing Author values, or click New to create one.
Image-(16 pixels)	File name of the 16 pixel image. NNMi supports .jpg, .jpeg, .gif. and .png file types. Click **Browse** to navigate to the image file to be used for this icon. **Tip**: As you browse for the image file, NNMi displays the image and its size (for example 16x16). **Note**: Ensure that the image background is transparent. After you select the image file, click **Open**.
Image-(32 pixels)	File name of the 32 pixel image. NNMi supports.jpg, ..jpeg, gif. and .png file types. Click **Browse** to navigate to the image file to be used for this icon. **Tip**: As you browse for the image file, NNMi displays the image and its size (for example 32x32).

Device Profile Icons, continued

Attribute	Description
Name	Enter a unique name that identifies the icon. Type a maximum of 64 characters. Alpha-numeric and special characters (~ ! @ # $ % ^ &amp; * ( ) _+ -) are permitted. Spaces are not permitted. **Note**: To enable you to filter the Icons table view by vendor, include the vendor name and the Device Profile attribute for which the icon will be used. For example, for a Device Family icon, you might use the following format: • <*unique_information*>-<*vendor_name*>-`family` For example, for a Device Vendor icon, you might use the following format: • <*unique_information*>-<*vendor_name*>-`vendor` For example, for a Device Category icon, you might use the following format: • <*unique_information*>-<*vendor_name*>- -`category` **Tip**: See the **Name** drop-down menu for example names of icons provided by HP.
	Note: Ensure that the image background is transparent. After you select the image file, click **Open**.

View the Device Profile Icons Available

NNMi enables you to customize the icons associated with a Device Profile or specific nodes. These icons appear in views, menu items, and as foreground images on an NNMi topology map.

Tip: To use the command line to load the icon images to the NNMi database so they are available for use, see nnmicons.ovpl. To use the NNMi console to load images, see "Add Device Profile Icons" on page 474

Note: NNMi provides a "missing" icon (🖼) to indicate a Device Profile icon is not available. Reasons NNMi displays a "missing" icon includes 1) the icon has been deleted or 2) the icon does not exist.

To view the device profile icons available for use:

Navigate to the **Icons** option under the **User Interface** folder

1. Navigate to the **Configuration** workspace.
2. Click to expand **User Interface**.
3. Select **Icons**.

For each icon, NNMi displays the image (in 16 pixels), the name of the image, and the author.

Note: Images must be provided in 16 pixels. You can also specify the same image in 32 pixels. To determine whether a 32 pixel image has been added to the NNMi database, examine the Analysis Pane information for the selected icon.

To modify or delete an icon using the command line, see nnmicons.ovpl.

To modify or delete an icon using the NNMi console, see "Customize Device Profile Icons" on page 474

Change the Image for a Specified Icon

NNMi enables you to customize the icons associated with a Device Profile or specific Nodes. These icons appear in views, menu items, and as foreground images on an NNMi topology map.

You can specify icons for a specific Node, Device Family, Device Vendor, or Device Category. NNMi determines the icon to use in the following order of precedence:

- Specified Node Icons
- Device Family Icons
- Device Vendor Icons
- Device Category Icons

Tip: To use the command line to load the icon images to the NNMi database so they are available for use, see nnmicons.ovpl. To use the NNMi console to load images, see "Add Device Profile Icons" on page 474

Note: NNMi provides a "missing" icon () to indicate a Device Profile icon is not available. Reasons NNMi displays a "missing" icon includes 1) the icon has been deleted or 2) the icon does not exist.

To change the image for a specified icon:

1. Navigate to the **Icons** option under the **User Interface** folder
   a. Navigate to the **Configuration** workspace.
   b. Click to expand **User Interface**.
   c. Select **Icons**.
2. Select the icon image that you want to change.
3. Provide settings for the selected icon. (see the Device Profile Icon Images table).
4. Click **Save and Close** to the **Icon** form.
5. Click **Save and Close** to save and apply your changes.

Device Profile Icon Images

Attribute	Description
Name	The unique name that identifies the icon.
Description	Provide additional information that you want to store about the icon. Type a maximum of 2048 characters. Alpha-numeric, spaces, colons (:), and special characters (~ ! @ # $ % ^ & * ( ) _+) are permitted.
Author	The name of the Author who created the icon object. See Author form for important information.

Device Profile Icon Images, continued

Attribute	Description
Name	The unique name that identifies the icon.
	Caution: It is recommended that you create new icon objects rather than modify icons provided by HP. If the Author attribute value is **HP Network Node Manager**, any changes are at risk of being overwritten in the future. Also see: "Export/Import Behavior and Dependencies" on page 1579 Click the Lookup icon and select Show Analysis to display details about the currently selected Author, select Quick Find to access the list of existing Author values, or click New to create one.
Image-(16 pixels)	File name of the 16 pixel image. NNMi supports .jpg, .jpeg, .gif. and .png file types. Click **Browse** to navigate to the image file to be used for this icon. **Tip**: As you browse for the image file, NNMi displays the image and its size (for example 16x16). **Note**: Ensure that the image background is transparent. After you select the image file, click **Open**.
Image-(32 pixels)	*Optional*. File name of the 32 pixel image. NNMi supports.jpg, ..jpeg, gif. and .png file types. Click **Browse** to navigate to the image file to be used for this icon. **Tip**: As you browse for the image file, NNMi displays the image and its size (for example 32x32). **Note**: Ensure that the image background is transparent. After you select the image file, click **Open**.

Configure the Device Profile Icon for Specified Nodes

NNMi enables you to customize the icons associated with a device profile.These icons appear in views, menu items, and as foreground images on an NNMi topology map.

You can specify image icons for a specific Node, Device Family, Device Vendor, or Device Category. NNMi determines the icon image to user per node in the following order of precedence:

- Specified Node Icons
- Device Family Icons
- Device Vendor Icons
- Device Category Icons

Tip: To use the command line to load the icon images to the NNMi database so they are available for use, see nnmicons.ovpl. To use the NNMi console to load images, see "Add Device Profile Icons" on page 474

Note: NNMi provides a "missing" icon () to indicate a Device Profile icon is not available. Reasons NNMi displays a "missing" icon includes 1) the icon has been deleted or 2) the icon does not exist.

To configure the Device Profile Icon for specified Nodes:

1. Navigate to the Nodes view (for example **Inventory** > **Nodes**).
2. Use CTRL-Click to select each node.

   Tip: You can also select Nodes from a map view.
3. Select **Actions > Custom Attributes > Add** .
4. In the **Name** drop-down menu, select **NNM_ICON**.
5. In the **Value** attribute, enter the Name of the icon you want to use for the selected nodes prepended with **NNM**.

   Note: The Value *must* begin with `NNM`: For example: `NNM:1400-procurve-vendor`.

   Tip: To view the device profile icons available, see "View the Device Profile Icons Available" on page 477

Configure Device Family Icons

NNMi enables you to customize the icons associated with a Device Profile or specific Nodes. These icons appear in views, menu items, and as foreground images on an NNMi topology map.

You can specify image icons for a specific Node, Device Family, Device Vendor, or Device Category. NNMi determines the icon image to user per node in the following order of precedence:

- Specified Node Icons
- Device Family Icons
- Device Vendor Icons
- Device Category Icons

Tip: To use the command line to load the icon images to the NNMi database so they are available for use, see nnmicons.ovpl. To use the NNMi console to load images, see "Add Device Profile Icons" on page 474

To configure a Device Family icon:

1. Navigate to the **Device Profile** option in the **Configuration** workspace.
   a. Navigate to the **Configuration** workspace..
   b. Select **Device Profiles**.
2. Navigate to the **Device Family** attribute.
3. Click the **Lookup** icon, and select **Open.**

4. Navigate to the **Icon** attribute.
5. Do one of the following:
   a. Select the Name of the icon you want to use.

   b. Click the **Lookup** icon, and select **New**. See "Add Device Profile Icons" on page 474 for more information.
6. Click **Save and Close** to save and apply your changes.
7. Verify that the icon specified displays in the appropriate places.

   Note: NNMi provides a "missing" icon () to indicate a Device Profile icon is not available. Reasons NNMi displays a "missing" icon includes 1) the icon has been deleted or 2) the icon does not exist.

Configure Device Vendor Icons

NNMi enables you to customize the icons associated with a Device Profile or specific Nodes. These icons appear in views, menu items, and as foreground images on an NNMi topology map.

You can specify image icons for a specific Node, Device Family, Device Vendor, or Device Category. NNMi determines the icon image to user per node in the following order of precedence:

- Specified Node Icons
- Device Family Icons
- Device Vendor Icons
- Device Category Icons

Tip: To use the command line to load the icon images to the NNMi database so they are available for use, see nnmicons.ovpl. To use the NNMi console to load images, see "Add Device Profile Icons" on page 474

To configure a Device Vendor icon:

1. Navigate to the **Device Profile** option in the **Configuration** workspace.

   a. Navigate to the **Configuration** workspace..

   b. Select **Device Profiles**.
2. Navigate to the **Device Vendor** attribute.
3. Click the Lookup icon, and select **Open**.
4. Navigate to the **Icon** attribute.
5. Do one of the following:
   a. Select the Name of the icon you want to use.

   b. Click the **Lookup** icon, and select **New**. See "Add Device Profile Icons" on page 474 for more information.
6. Select the Name of the icon you want to use.
7. Click **Save and Close** to save and apply your changes.

8. Verify that the icon specified displays in the appropriate places.

   Note: NNMi provides a "missing" icon () to indicate a Device Profile icon is not available. Reasons NNMi displays a "missing" icon includes 1) the icon has been deleted or 2) the icon does not exist.

Configure Device Category Icons

NNMi enables you to customize the icons associated with a Device Profile or specific Nodes.These icons appear in views, menu items, and as foreground images on an NNMi topology map.

You can specify icons for a specific Node, Device Family, Device Vendor, or Device Category. NNMi determines the icon to use in the following order of precedence:

- "Configure the Device Profile Icon for Specified Nodes" on page 479
- "Configure Device Family Icons" on page 480
- "Configure Device Vendor Icons" on the previous page

Tip: To use the command line to load the icon images to the NNMi database so they are available for use, see nnmicons.ovpl. To use the NNMi console to load images, see "Add Device Profile Icons" on page 474

To configure a Device Category icon:

1. Navigate to the **Device Profile** option in the **Configuration** workspace.
   a. Navigate to the **Configuration** workspace..
   b. Select **Device Profiles**.
2. Navigate to the **Device Category** attribute.
3. Click the Lookup icon, and select **Open**.
4. Navigate to the **Icon** attribute.
5. Do one of the following:
   a. Select the Name of the icon you want to use.
   b. Click the **Lookup** icon, and select **New**. See "Add Device Profile Icons" on page 474 for more information.
6. Select the Name of the icon you want to use.
7. Click **Save and Close** to save and apply your changes.
8. Verify that the icon specified displays in the appropriate places.

   Note: NNMi provides a "missing" icon () to indicate a Device Profile icon is not available. Reasons NNMi displays a "missing" icon includes 1) the icon has been deleted or 2) the icon does not exist.

Customize Object Attributes

NNMi enables you to customize the Custom Attributes associated with a node or interface.

- "Add a Custom Attribute to a Node or Interface Object" below
- "Add Custom Attributes to Multiple Nodes or Interfaces" on the next page

Add a Custom Attribute to a Node or Interface Object

If you determine that you want to keep track of additional information about a node or interface, you can add Custom Attributes to these objects. For example, you might determine that you want to track the owner of your nodes on the network. You might also want to track the serial number for each node.

To add Custom Attributes to a Node object:

1. Navigate to the **Custom Attributes** tab:
   a. From the workspace navigation panel, select a workspace that contains a Node view. For example, the **Inventory** workspace.

   b. Double-click the row representing the node with settings you want to edit.

      Tip: You can also select Nodes from a map view.

   c. Select the **Custom Attributes** tab.
2. Click the ✱ New icon to create a Custom Attribute.
3. Enter a Name and Value. See Node Custom Attributes Form for more information.

   NNM iSPI Performance for Metrics*only*. You can use Custom Attributes to include additional Node or Interface information in NNM iSPI Performance for Metrics reports. Click here for more information.

   Create the required Custom Attribute using **Actions** > **Custom Attribute** > **Add**.

   Note: The Custom Attribute Name must be **NPS Annotation**. NPS (Network Performance Server) is the database server installed with the NNM iSPI Performance for Metrics software.

   Also see "Annotate NNM iSPI Performance for Metrics Reports" on page 1486.

   You can also use Custom Attributes to customize the Device Profile icon for one or more nodes. Click here for more information.

   To customize the Device Profile for one or more nodes:

   In the **Name** drop-down menu, select **NNM_ICON**.

   In the **Value** attribute, enter the name of the icon you want to use for the selected nodes.

   Tip: To view the device profile icons available, see "View the Device Profile Icons Available" on page 477
4. Click **Save and Close** to return to the main Node Form.
5. Click **Save and Close** to save your changes**.**

To add Custom Attributes to an interface object:

1. Navigate to the **Custom Attributes** tab:

   a. From the workspace navigation panel, select a workspace that contains an Interfaces view. For example, the **Inventory** workspace.

   b. Double-click the row representing the interface with settings you want to edit.

      Tip: You can also select Interfaces from a map view.

   c. Select the **Custom Attributes** tab.

2. Click the ✳ New icon to create a Custom Attribute.

3. Enter a Name and Value. See Interface Custom Attributes Form for more information.

   NNM iSPI Performance for Metrics*only*. You can use Custom Attributes to include additional Node or Interface information in NNM iSPI Performance for Metrics reports. Click here for more information.

   For example, you might want to add information that identifies the interface Wide Area Network circuit.

   Create the required Custom Attribute using **Actions** > **Custom Attribute** > **Add**.

   Note: The Custom Attribute Name must be **NPS Annotation**.

   Also see "Annotate NNM iSPI Performance for Metrics Reports" on page 1486.

4. Click **Save and Close** to return to the main Interface Form.

5. Click **Save and Close** to save your changes.

Related Topics

"Add Custom Attributes to Multiple Nodes or Interfaces" below

Add Custom Attributes to Multiple Nodes or Interfaces

Custom attributes can be added to multiple nodes or interfaces in one of two ways:

- Use the **Actions** → **Custom Attributes** option. See "Add Custom Attributes to Multiple Nodes or Interfaces Using the Actions Menu" below
- Use the nnmloadattributes.ovpl command line tool to add Custom Attributes to multiple Nodes. See "Add Custom Attributes to Multiple Nodes or Interfaces Using the Command Line" on page 486

Add Custom Attributes to Multiple Nodes or Interfaces Using the Actions Menu

NNM iSPI Performance for Metrics only. You can use Custom Attributes to include additional Node or Interface information in NNM iSPI Performance for Metrics reports. Click here for more information.

For example, you might want to add information that identifies the interface Wide Area Network circuit:

- Name must be set to **NPS Annotation**. NPS (Network Performance Server) is the database server installed with the NNM iSPI Performance for Metrics software.
- Value can be any text that identifies the interface's Wide Area Network circuit.

Also see "Annotate NNM iSPI Performance for Metrics Reports" on page 1486.

You can use Custom Attributes to customize the Device Profile icon for one or more nodes. Click here for more information.

- Name must be set to **NNM_ICON**.
- Value attribute is the name of the icon you want to use for the selected nodes. To view the list of available device profile icons, see "View the Device Profile Icons Available" on page 477.

To add a Custom Attribute to multiple nodes or interfaces using the Actions menu (if your role permits you to do this):

1. Navigate to a **Node** or **Interface** inventory view.
   a. From the workspace navigation panel, select the **Inventory** workspace.
   b. Select the node or interface view of interest (or example, **Nodes** view).

   > **Tip:** You can also select Nodes or Interfaces from a map view.

2. Use Ctrl-Click to select each node or interface to which you want to add to a Custom Attribute.
3. Select **Actions** → **Custom Attributes** → **Add**.
4. In the **Custom Attributes** dialog, box, enter the following:

Name	Do one of the following: ▪ Select a previously established entry from the drop-down list. For example: ○ NPS Annotation (to enhance NNM iSPI Performance for Metrics reports, see "Annotate NNM iSPI Performance for Metrics Reports" on page 1486) ○ NNM_ICON (to customize the Device Profile icon) ▪ Type any value directly into the drop-down list to created a new one: Maximum of 255 characters. Alpha-numeric, spaces, and special characters (~ ! @ # $ % ^ & * ( ) _ + -)
Value	Type the value you want to assign to the Custom Attribute for the selected node: Maximum of 255 characters. Alpha-numeric, spaces, and special characters (~ ! @ # $ % ^ & * ( ) _ + -)

5. Click **OK**.

   The value appears in the table view on the Node form's Custom Attributes tab. For more information, see Node Form: Custom Attributes Tab.

Add Custom Attributes to Multiple Nodes or Interfaces Using the Command Line

The nnmloadattributes.ovpl command line tool enables you to load Custom Attributes from a comma-separated values (CVS) file. This feature is useful if you have information about a large number of nodes or interfaces defined in an external data storage, and you would like to load that information into the NNMi database as Custom Attributes. For example:

- Node location information in a Microsoft Excel spreadsheet where you track the location of each node: You can save this information as a .csv file. Use the `nnmloadattributes.ovpl` command to define **BldgLocation** as a Custom Attribute and load the location values for each node into the NNMi database. You can then create a Node Group with an Additional Filters specification using **BldgLocation** as the `customAttrName` and the location of interest, such as **Building Five Upper** as the `customAttrValue`.
- Interface information in a comma-separated value file where you track the name of customers assigned to each interface: Use the `nnmloadattributes.ovpl` command to define **Customer** as a Custom Attribute and load the name values for each customer into the NNMi database. You can then create an Interface Group with an Additional Filters specification using **Customer** as the `customAttrName` and a customer name, such as **Hewlett Packard** as the `customAttrValue`.
- *HP Network Node Manager iSPI Performance for Metrics Software only*. You can use Custom Attributes to include additional Node or Interface information in NNM iSPI Performance for Metrics reports. Create the required Custom Attribute using the **Actions** > **Custom Attributes** > **Add**. The Custom Attribute Name must be **NPS Annotation**. See "Annotate NNM iSPI Performance for Metrics Reports" on page 1486 for more information.

To load Custom Attributes for Nodes or Interfaces using a comma-separated file:

See the nnmloadattributes.ovpl Reference Page for more information about the `nnmloadattributes.ovpl` command, including requirements for the CSV file. You must provide a CSV file with a specific syntax and order. Each column in the CSV file has a pre-defined meaning.

```
nnmloadattributes.ovpl -u <NNMiadminUsername> -p <NNMiadminPassword> -
r [true|false] -t node -f <CSV file name>
```

`-r true` = the value of any existing Custom Attribute with the same `customAttrName` is overwritten with the value in your CSV file. The default setting is `-r false` = if the `customAttrName` already exists, the `nnmloadattributes.ovpl` command does not change the previous `customAttrValue`.

`-t` is used to specify the object type on which the attributes should be loaded. Use `node` to load Custom Attribute information for nodes.

CSV file name is the name of the CSV file that contains the Node Custom Attribute information.

Remove Custom Attributes from Multiple Nodes or Interfaces Using the Actions Menu

To remove a Custom Attribute from multiple nodes or interfaces using the Actions menu (if your role permits you to do this):

1. Navigate to a **Node** or **Interface** inventory view.
   a. From the workspace navigation panel, select the **Inventory** workspace.
   b. Select the node or interface view of interest (or example, **Nodes** view).

      Tip: You can also select Nodes or Interfaces from a map view.
2. Use CTRL-Click to select each node or interface to which you want to add to a Custom Attribute.
3. Select **Actions** → **Custom Attributes** → **Remove**.
4. Click **OK**.

Configure Maps

NNMi enables you to configure the following maps:

- Node Group Map views
- Path View Maps

  Note: The Node Group Overview map provided by NNMi is not configurable.

When configuring Node Group maps, you can do the following:

- Include only the nodes that are important to you.
- Specify which Node Group maps appear in the Topology Maps workspace.
- Specify refresh information.
- View node groups in the context of a relevant background image, such as a map illustrating node locations.
- View node groups in a customized arrangement.

When configuring Node Group map views, you can also specify the role level required to save maps in a customized arrangement. See "Define Node Group Map Settings" on the next page for more information.

When configuring Path View maps you specify undiscovered regions of your network by creating a `PathConnections.xml` file that defines the path between the undiscovered nodes. See"Configure a Path View Map" on page 498 for more information.

You can also specify the maximum number of nodes to display on a map. See "Define Default Map Settings" on page 470 for more information.

Related Topics

"Node Group Map Settings Form" on the next page

Node Group Map View

Position Nodes in a Node Group Map

Define Node Group Map Settings

Node Group Map settings specify the node group and background image to be used in a Node Group map. Map settings include the following:

- Node group name
- Topology Maps Workspace ordering
- Minimum role for saving edited locations for each node in the map
- Refresh interval
- The maximum number of map nodes
- Node connectivity information
- Node Group connectivity information
- Background image information

Node Group Map views are used for a variety of purposes in NNMi:

- Viewing groups of only the nodes that are important to you.
- Viewing Node Groups in the context of a relevant background image.
- Viewing Node Groups in a customized arrangement.

To define Node Group Map Settings, use the "Node Group Map Settings Form" below.

To view a Node Group Map, use the **Actions** menu from the NNMi main toolbar from either a Node Group or Node Group Map Settings. See Node Group Map for more information.

Tip: You can right-click any object in a table or map view to access the **Actions** menu.

To view more information about the Node Group from a Node Group map, use the **File** → **Open Node Group for Map** option to open the Node Group form for the selected Node Group.

Node Group Map Settings Form

Use the Node Group Map Settings form to configure maps based on currently defined Node Groups. Items you configure include the background image and type of connectivity (for example, Layer 2) to be displayed on the map.

Note: NNMi displays the list of Node Group Map Settings that have default configuration changes. If NNMi does not display a list of Node Group Map Settings, this means that NNMi is using the default settings for each Node Group Map. To change the default settings for a Node Group Map, either reposition the nodes on the map of interest and select **Save Layout** from the Node Group Map toolbar or use the Node Group Map Settings form to create a Node Group Map Settings configuration as described below. See Position Nodes on a Node Group Map for

> more information about using **Save Layout**.

To configure Node Group Map Settings, do the following:

1. Navigate to the **Node Group Map Settings** view.

   > **Note:** You can also access the Node Group Map Settings form from any Node Group Map by using the **File** → **Open Node Group Map Settings** option.

   a. From the workspace navigation panel, select the **Configuration** workspace.

   b. Click to expand **User Interface**.

   c. Select **Node Group Map Settings**.

   d. Do one of the following:

      - To create a new configuration, click the ✱ New icon.
      - To edit an existing configuration, double-click the row representing the Node Group Map Settings definition you want to edit.

2. Make your configuration choices (see table).

3. Click **Save and Close** to save and apply your changes.

Tasks for Configuring Node Group Map Settings

Task	How
"Configure Basic Settings for a Node Group Map" below	Use the Basics Settings pane to configure Node Group, Topology Maps, and Refresh Interval information. **Note**: To apply your Topology Maps Ordering configuration changes, sign out of the NNMi console. After restarting the console, your changes should take affect. Possible configuration changes include reordering a Node Group map view or adding a new Node Group map view to the workspace.
"Configure the Connectivity to be Displayed for a Node Group Map" on page 493	Use the Connectivity tab to configure the level of node connectivity to be displayed on the Node Group Map. Use this tab to also specify the Node Group connectivity to be displayed and maximum connections to be included on the Node Group map.
"Configure Background Image Information for a Node Group Map" on page 494	Use the Background Image tab to configure information about the Background Image to use on the Node Group map.

Configure Basic Settings for a Node Group Map

The Basic Settings configuration determines general information about the Node Group map.

To establish Basic Settings for a Node Group Map:

1. Navigate to the **Node Group Map Settings** view.

   > **Note:** You can also access the Node Group Map Settings form from any Node Group Map by using the **File** → **Open Node Group Map Settings** option.

   a. From the workspace navigation panel, select the **Configuration** workspace.

   b. Click to expand **User Interface**.

   c. Select **Node Group Map Settings** .

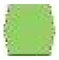
   d. Do one of the following:

      - To create a Node Group Map Settings definition, click the ✳ New icon.
      - To edit a Node Group Map Settings definition, double-click the row that represents the Node Group Map Settings definition you want to edit.
      - To delete a Node Group Map Settings definition, select a row and click the ✖ Delete button.

2. Establish the appropriate settings to identify Node Group and Refresh Settings information (see tables).

3. Click **Save and Close** to save and apply your changes.

Basic Attributes

Attribute	Description
Node Group	Specifies which parent node group to display in the Node Group Map view. The contents of the parent node group include any nodes and Child Node Groups associated with it. **Note**: NNMi displays any Child Node Groups of the selected parent Node Group as a hexagon on the map. The **Expand Child in Parent Node Group Map** attribute determines how a Child Node Group appears on the Node Group Map. **Expand Child in Parent Node Group Map** is disabled by default. • If the Child Node Group has the **Expand Child in Parent Node Group Map** attribute *disabled*, the Child Node Group appears as a hexagon on the map as shown below:  • If any Child Node Group has the **Expand Child in Parent Node Group Map** attribute *enabled*, NNMi instead recursively displays each of the nodes in that Child Node Group on the map.

Basic Attributes, continued

Attribute	Description
	See Node Group Form: Child Node Groups Tab for more information about configuring Child Node Groups.
Topology Maps Ordering	Use this attribute to specify the order in which you want the Node Group map to appear in the **Topology Maps** workspace. **Note**: If you do not want this Node Group map to appear in the **Topology Maps** workspace, leave the value blank. See Views Provided by NNMi for more information about the maps provided in the **Topology Maps** workspace. **Note**: To apply your Topology Maps Ordering configuration changes, sign out of the NNMi console. After restarting the console, your changes should take affect. Possible configuration changes include reordering a Node Group map view or adding a new Node Group map view to the Topology Maps workspace.
Minimum NNMi Role to Save Layout	Controls the minimum NNMi User Group required to save the layout of repositioned nodes in a Node Group Map. This value also controls the minimum User Group for configuring Node Group Map Settings. **Note**: Only a User Account assigned to the NNMi **Administrators** User Group can set the Minimum NNMi Role for Saving Map Layout value. Possible values include: • Administrator • Operator Level 2 • Operator Level 1 (with less access privileges than Level 2) The default value is *Administrator.* See "Determine which NNMi User Group to Assign" on page 549 for more information about NNMi roles. **Note**: A user with any NNMi Role can initially reposition nodes on a Node Group Map view. However, unless your user name is assigned to the required minimum NNMi Role, you cannot save the new node locations on the map. After being saved, these node positions are seen by any user opening this Node Group Map.
Map Refresh Interval	Specify the Refresh Interval you want to use in days, hours, minutes, and seconds. By default, the Refresh Interval is 30 seconds. This interval is used to set the Refresh Status interval for this map if it is used.
Maximum Number of Displayed Nodes	Specifies the maximum number of nodes to be displayed on the Node Group map. **Note**: This number applies to the total number of nodes within the Node Group, including the nodes in any Child Node Groups displayed on the map.
Maximum	Specifies the maximum number of end points to be displayed on a map.

Basic Attributes, continued

Attribute	Description
Number of Displayed End Points	**Note**: If maps are taking longer than expected to display, you might need to re-adjust this number.
Multiconnection Threshold	Use this attribute to change the number of connections that must exist between two Node Groups before NNMi displays the connections as one thick line (known as a multiconnection) on a Node Group map. Note the following: • The value you enter overrides the Multiconnection Threshold set using the Default Map Settings. • If this setting is blank, NNMi uses the Multiconnection Threshold value configured in Default Map Settings. • When this number of connections is reached, NNMi displays the connections as one thick line. • To display the Interface objects and each connection, double-click the line representing the multiconnection.
Indicate Key Incidents	In the Node Group map, NNMi can enlarge the map symbol of any node associated with a **Key Incident**[1]. Uers can click the **Indicate Key Incidents** button in the map view toolbar to toggle this feature on and off (see Using the View Toolbars: Node Group Map Toolbar Icons): 🔑 (on) = When the this Node Group map opens, NNMi enlarges any objects on a Node Group map that are Source Objects for a **Key Incident**[2]. (For example, when viewing the Node Group map, NNMi enlarges any node on a Node Group map that has an open root cause incident associated with it.) 🔑 (off) = When the this Node Group map opens, NNMi does not indicate the objects on a Node Group map that are Source Objects for a **Key Incident**[3].
Include in Visio Export	When ☑ enabled, NNMi includes this map when exporting all saved Node Group maps using the **Tools → Visio Export (iSPI NET only)→ Saved Node Group Maps** option.

[1] Incidents with both: (1) Severity = other than Normal. (2) Correlation Nature = equal to Root Cause, Service Impact, Stream Correlation, Rate Stream Correlation, Info, or None.
[2] Incidents with both: (1) Severity = other than Normal. (2) Correlation Nature = equal to Root Cause, Service Impact, Stream Correlation, Rate Stream Correlation, Info, or None.
[3] Incidents with both: (1) Severity = other than Normal. (2) Correlation Nature = equal to Root Cause, Service Impact, Stream Correlation, Rate Stream Correlation, Info, or None.

Basic Attributes, continued

Attribute	Description
	When □ disabled, NNMi does not include this map when exporting all saved Node Group maps using the **Tools** → **Visio Export (iSPI NET only)** → **Saved Node Group Maps** option.

Configure the Connectivity to be Displayed for a Node Group Map

The Connectivity Tab of the Node Group Map Settings form enables you to specify the level of connectivity to be displayed on the Node Group map. You also specify the connections that you want to display.

1. Navigate to the **Connectivity tab** of the **Node Group Map Settings** form.

   > **Note:** You can also access the Node Group Map Settings form from any Node Group Map by using the **File** → **Open Node Group Map Settings** option.

   a. From the workspace navigation panel, select the **Configuration** workspace.

   b. Click to expand **User Interface**.

   c. Select **Node Group Map Settings**.

   d. Do one of the following:

      ○ To create a Node Group Map Settings definition, click the ✱ New icon.

      ○ To edit a Node Group Map Settings definition, double-click the row representing the Node Group Map Settings definition you want to edit.

      ○ To delete a Node Group Map Settings definition, select a row and click the ✖ Delete button

   e. Navigate to the **Connectivity** tab.

2. Configure the connectivity information for this Node Group Map Settings definition (see table).

3. Click **Save and Close** to save and apply your changes.

Connectivity Attributes

Attribute	Description
Connectivity Type	Connectivity Type determines the type of connectivity to display between nodes in the Node Group Map view. By default, NNMi displays the Layer 2 connectivity between nodes when displaying a Node Group Map view. Possible values include: • **None** - Choose this if you do not want any connectivity displayed on the map. • **Layer 2** - Uses Layer 2 connectivity when displaying devices in a Node Group Map view. This connectivity is used by default when positioning node locations on a Node Group Map.

Connectivity Attributes, continued

Attribute	Description
	- **Layer 3** - Uses Layer 3 connectivity when displaying devices on a Node Group Map view. See Position Nodes on a Node Group Map for more information.
Add L2 Subnet Connections	If you specify **Layer 3** or **None** as the Connectivity Type, this option specifies that you want to include any subnet connections determined by IPv4 Subnet Connections Rules. See "Configure IPv4 Subnet Connection Rules" on page 244 for more information.
Add L2 User Connection Edits	If you specify **Layer 3** or **None** as the Connectivity Type, specifies that you want to include any Layer 2 Connections added using the NNMi nnmconnedit.ovpl command to add or delete connection data. See "Add or Delete a Layer 2 Connection" on page 284 for more information.
Interface Group	Use this option, if you want to reduce the connectivity endpoints on the Node Group Map. The Interface Group you select defines the Interface Group to which an interface must belong to be used to connect a Node Group to a Node Group or a Node to a Node Group. NNMi displays Layer 2 endpoints that are interfaces in the group. NNMi displays Layer 3 endpoints that are IP addresses associated with interfaces in the group.
Nodes to Node Group	Select this check box if you want Node to Node Group connectivity to appear on the Node Group map. **Note**: By default, this option is not enabled.
Node Groups to Node Groups	Select this check box if you want Node Group to Node Group connectivity to appear on the Node Group map. **Note**: By default, this option is not enabled.

Configure Background Image Information for a Node Group Map

Use the Background Image tab of the Node Group Map Settings form to configure information about the Background Image to use on the Node Group map.

1. Navigate to the **Background Image** tab of the **Node Group Map Settings** form.

   > **Note:** You can also access the Node Group Map Settings form from any Node Group Map by using the **File → Open Node Group Map Settings** option.

   a. From the workspace navigation panel, select the **Configuration** workspace.

   b. Click to expand **User Interface**.

c. Select **Node Group Map Settings**.

d. Do one of the following:

- To create a Node Group Map Settings definition, click the ✳ New icon.
- To edit a Node Group Map Settings definition, double-click the row representing the Node Group Map Settings definition you want to edit.
- To delete a Node Group Map Settings definition, select a row and click the ✖ Delete button.

2. Establish the appropriate settings to identify the Background Image information (see table).

3. Click **Save and Close** to save and apply your changes.

Background Image Attributes

Attribute	Description

Background Image	Enter the URL for the background image you want to use for this Node Group Map. You can use a background image provided by NNMi or add your own.
	Note: Click **Background Image** to view the map.
	Use a Background Image Provided by NNMi
	NNMi provides a set of background images that include maps of many countries. If you want to use one of those images, append the location and file name to the URL at which you access the NNMi console. Use the format: `/nnmbg/<file name>`. For example:
	`/nnmbg/colorado.gif`
	To see all of the available images provided by NNMi, browse to:
	`http://`*`<serverName>`*`:`*`<portNumber>`*`/nnmbg/`
	Note: If the NNMi Web server uses the https protocol, use `https` instead of `http`. See the "Working with Certificates for NNMi" chapter in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.
	`<serverName>` = the fully-qualified domain name of the NNMi management server (values allowed here are determined by the *Enable URL Redirect* setting in User Interface Configuration, see "Configuring the NNMi User Interface" on page 467)
	`<portNumber>` = the NNMi HTTP port number
	Use a Background Image You Provide
	You can also provide your own images. See "Background Image Sources in Node Group Maps" on the next page for more information about where to load the background images you want to use.
	To see a list of all the images added to NNMi, access the following URL:

Background Image Attributes, continued

Attribute	Description

	`http://<serverName>:<portNumber>/nnmdocs/images/`
	To use an image that has been added to NNMi, use the following URL:
	`/nnmdocs/images/<file name>`
	For example: `/nnmdocs/images/myimage.gif`
	Note the following:
	• NNMi accepts images that can be loaded by a Web browser. Common file extensions include: .gif, .png and .jpg.
	• Image names are case sensitive. All background image file names provided by NNMi are lowercase.
	• Do not use `http://<localhost>` in your URL. This implies the image is on your local machine and is not available from other clients.
	• If using full URLs, all client machines must be able to resolve the DNS hostname of the server on which the images reside.
	• When you pan and zoom around the map, the background image moves in relation with the other objects on the map.
	If the image does not display, see "Troubleshoot URLs When Specifying a Background Image" on page 498 for more information.

Background Image Scale	The Background Image Scale attribute applies to the actual background image dimensions when displayed on a Node Group Map. Enter a floating point number greater than zero (0.0) to indicate the ratio at which you want NNMi to scale the background image. For example, the value 1.0 represents a one-to-one ratio, resulting in a background image displayed at actual size. A value of 2.0 represents a two-to-one ratio, resulting in a background image displayed at twice the actual size. **Note**: The default ratio value is 1.0. (This means no scaling is applied.) Use this default value initially. You can adjust it as needed based on the relative size between the image and nodes. See "Scale Background Images in Node Group Maps" on the next page for guidelines for scaling the background images you specify.

Background Image Sources in Node Group Maps

When specifying background images to include in Node Group Maps, NNMi enables you to use images provided by NNMi or images that you provide.

The images that NNMi provides include maps of many countries.

To see the available images provided by NNMi:

Browse to: `http://<serverName>:<portNumber>/nnmbg/`

> **Note:** If the NNMi Web server uses the https protocol, use `https` instead of `http`. See the "Working with Certificates for NNMi" chapter in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.

<serverName> = the fully-qualified domain name of the NNMi management server (values allowed here are determined by the *Enable URL Redirect* setting in User Interface Configuration, see "Configuring the NNMi User Interface" on page 467)

<portNumber> = the NNMi HTTP port number

To use your own background images:

Place you user-supplied images in the following directory:

Windows:

`%NnmDataDir%/shared/nnm/www/htdocs/images`

Unix:

`/var/opt/OV/shared/nnm/www/htdocs/images`

NNMi accepts images that can be loaded by a Web browser. Common file extensions include: .gif, .png and .jpg.

To see the available images that have been added to NNMi:

Access the following URL: `http://<serverName>:<portNumber>/nnmdocs/images`

> **Note:** If the NNMi Web server uses the https protocol, use `https` instead of `http`. See the "Working with Certificates for NNMi" chapter in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.

<serverName> = the fully-qualified domain name of the NNMi management server (values allowed here are determined by the *Enable URL Redirect* setting in User Interface Configuration, see "Configuring the NNMi User Interface" on page 467)

<portNumber> = the NNMi HTTP port number

See "Node Group Map Settings Form" on page 488 for more information about how to configure Node Group Maps to use background images.

Scale Background Images in Node Group Maps

Scale a specified background image for a Node Group Map using the Background Image Scale attribute. See "Define Node Group Map Settings" on page 488 for more information.

When you use the maps provided by NNMi, it is recommended that you initially use the default value of 1.0 for the Background Image Scale.

When you use your own images for map backgrounds and you are selecting a scale value, consider the following:

- NNMi renders its nodes 50 by 50 pixels. This means if your image is 500 pixels wide, there is room for 10 nodes across the image.
- To display the image at normal resolution, enter a scale value of 1.0. (This means no scaling occurs.)
- After the image displays on the map, look at the relationship between the node size and the background to determine whether you need to rescale the background image:
  - If the nodes look too large compared to the background, enlarge the image using a scale value greater than 1.0.
  - If the nodes look too small compared to the background, make the image smaller using a scale value less than 1.0.

Troubleshoot URLs When Specifying a Background Image

This topic contains troubleshooting steps to use if your background image does not display.

If you used a relative URL (beginning with a slash (/) in the Background Image attribute value:

1. Copy and paste the URL to a browser.
2. Insert `http://<serverName>:<portNumber>` in front of the slash (/).

> **Note:** If the NNMi Web server uses the https protocol, use `https` instead of `http`. See the "Working with Certificates for NNMi" chapter in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.

`<serverName>` = the fully-qualified domain name of the NNMi management server (values allowed here are determined by the *Enable URL Redirect* setting in User Interface Configuration, see "Configuring the NNMi User Interface" on page 467)

`<portNumber>` = the NNMi HTTP port number

If you used an absolute URL (beginning with http://) in the Background Image attribute value:

Copy and paste the URL to a browser.

Configure a Path View Map

Configuring a Path View map is useful when you have two or more areas of your network which are separated by undiscovered devices, such as service provider nodes. NNMi enables you to configure a Path View map that traverses undiscovered regions of your network. To configure this kind of Path View map, create a `PathConnections.xml` file that defines the following:

- Required. A Start node for each `<CONNECT>` to be included in the Path View map

  Note: The Start node specified must be a Router or Switch-Router device that is managed by NNMi.
- *Optional*. A unique identifier for a `<CONNECT>`

- *Optional*. The outbound interface from each Start node per `<CONNECT>`
- Required. Any number of undiscovered nodes you want to be included in the map between each `<CONNECT>`
- *Optional.* An End node for a `<CONNECT>` to be included in the Path View map.

  Note: The End node specified must be a Router or Switch-Router device that is managed by NNMi.
- *Optional*. The inbound interface to each End node per `<CONNECT>` specified.

Each time NNMi determines a node in the Path View, NNMi checks whether the node is specified as a Start node in the `PathConnections.xml` file. If the node is specified as a Start node in `PathConnections.xml`, each `<CONNECT>` configured in `PathConnections.xml` is inserted in the Path View map.

Note: *NNMi Advanced*. NNMi can use RAMS data to determine router paths. When RAMS data is used to determine the router paths, NNMi ignores the `PathConnections.xml` file. See Path View with NNMi Advanced for more information.

Note: *NNMi Advanced*. Path View works only with IPv4 addresses. The NNMi Advanced IPv6 address values are not valid choices for Path View. Any devices in your network that are configured with IPv6 addresses cannot be displayed on Path View maps.

To configure a Path View map:

Using the required format, create a `PathConnections.xml` file in the following location:

Windows:
`%NnmDataDir%/shared/nnm/conf/PathConnections.xml`

UNIX:
`/var/opt/OV/shared/nnm/conf/PathConnections.xml`

The following table describes each of the file elements and its format requirements. (Also see the sample file)

Note: Each segment of the path that you specify using the `<CONNECT>` element is directional. If you want to view the path between two nodes in both directions, make sure you include the Start and End nodes for each direction. You should also include the inbound interface for the Start node. If you do not limit the possible routers by including the inbound interface for the Start node, Path View might find additional routers in the path.

Elements for the Path View Configuration File

Element Descriptions
`<CONNECTIONS>` Required parent element. The file must include only one `<CONNECTIONS>` element.
`<CONNECT>` Specifies a segment of the path. Each `<CONNECT>` designates a start and stop location for the `<CONNECT>`. The file can include more than one `<CONNECT>` element.

Elements for the Path View Configuration File, continued

Element Descriptions

`<ID>`
`    C1`
`</ID>`
Optional. Identifies the connection. NNMi uses the ID value you enter when reporting errors for a `<CONNECT>`.
If you do not provide an ID value for the path between a Start and End node, any error message for the `<CONNECT>` displays `Not Applicable` rather than the unique identification value.

`<START>`
`    <IP_OR_DNS>xxx.xx.xxx.x</IP_OR_DNS>`
`     <OUTBOUND_INTERFACE_IFINDEX>x</OUTBOUND_INTERFACE_IFINDEX>`
`     <NEXT_HOPS>`
`          <HOP>xxx.xx.xxx.x</HOP>`
`          <HOP>xxx.xx.xxx.x</HOP>`
`     </NEXT_HOPS>`
`</START>`
Specifies the node where a segment of the path starts. You provide values for the following elements:
• `<IP_OR_DNS>` provides the name or IPv4 address of a node in your network. See "Configure the Node Name Strategy" on page 203 for more information about node names.
• *Optional*. `<OUTBOUND_INTERFACE_IFINDEX>` designates which of the Start node's interfaces to use for this segment of the path.
• `<NEXT_HOPS>` designates one or more specific IPv4 addresses or nodes that you want to be included in the path.

`<END>`
`  <IP_OR_DNS>xxx.xx.xxx.x</IP_OR_DNS>`
`  <INBOUND_INTERFACE_IFINDEX>x</INBOUND_INTERFACE_IFINDEX>`
`</END>`
Specifies the node where the `<CONNECT>` ends. You provide values for the following elements:
• `<IP_OR_DNS>` provides the name or IPv4 address of a node in your network.
• *Optional.*`<INBOUND_INTERFACE_IFINDEX>` designates which of the End node's interfaces to use for this segment of the path.

`</CONNECT>` Required. Designates the end of the XML code that defines one segment of your path view.
`</CONNECTIONS>` Required parent element. Designates the end of the XML code that defines your path view.

Click here to view a sample file:

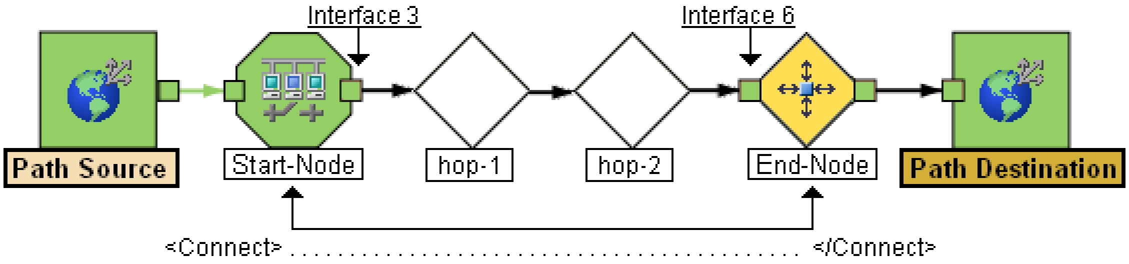
```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<CONNECTIONS>
     <CONNECT>
          <ID>
             C1
          </ID>
          <START>
                <IP_OR_DNS>StartNode.xx.xxx.x</IP_OR_DNS>
             <OUTBOUND_INTERFACE_IFINDEX>3</OUTBOUND_INTERFACE_
IFINDEX>
                <NEXT_HOPS>
                     <HOP>hop-1.xxx.xx.xxx</HOP>
                    <HOP>hop-2.xxx.xx.xxx</HOP>
                </NEXT_HOPS>
          </START>
          <END>
                <IP_OR_DNS>EndNode.xxx.xx.xxx</IP_OR_DNS>
                <INBOUND_INTERFACE_IFINDEX>6</INBOUND_INTERFACE_
IFINDEX>
          </END>
     </CONNECT>
</CONNECTIONS>
```

When viewing Path View maps that are configured using the `PathConnections.xml` file, note the following:

- If the `<END>` element is not specified, NNMi connects directly to the Destination node to complete the path.
- If the `<END>` element is specified, then the associated `<IP_OR_DNS>` specifies a discovered node as the End node of this segment of your Path View.

Click here to view the sample Path View map generated from the sample file above.



Click here to view a sample file that includes both directions for the sample Path View map above.

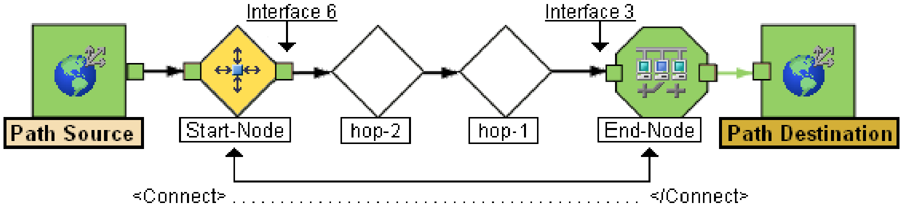
Note: In this example, the path is the same in both directions. In many cases, the path might be different in each direction.

```
<?xml version="1.0" encoding="UTF-8"?>

<CONNECTIONS>
   <CONNECT>
        <ID>
           C1
        </ID>
          <START>
                <IP_OR_DNS>StartNode.xx.xxx.x</IP_OR_DNS>
```

```
                    <OUTBOUND_INTERFACE_IFINDEX>6</OUTBOUND_INTERFACE_
IFINDEX>
                <NEXT_HOPS>
                     <HOP>hop-1.xxx.xx.xxx</HOP>
                    <HOP>hop-2.xxx.xx.xxx</HOP>
                </NEXT_HOPS>
          </START>
          <END>
                <IP_OR_DNS>EndNode.xxx.xx.xxx</IP_OR_DNS>
                <INBOUND_INTERFACE_IFINDEX>3</INBOUND_INTERFACE_
IFINDEX>
          </END>
     </CONNECT>
</CONNECTIONS>
```

Click here to view the sample Path View map generated from the sample file above after clicking the **Swap Nodes** button.



Configure Menus

As an NNMi administrator, you configure how menu items are nested in the NNMi console. See "Create Menu Nesting" on page 1415 for more information.

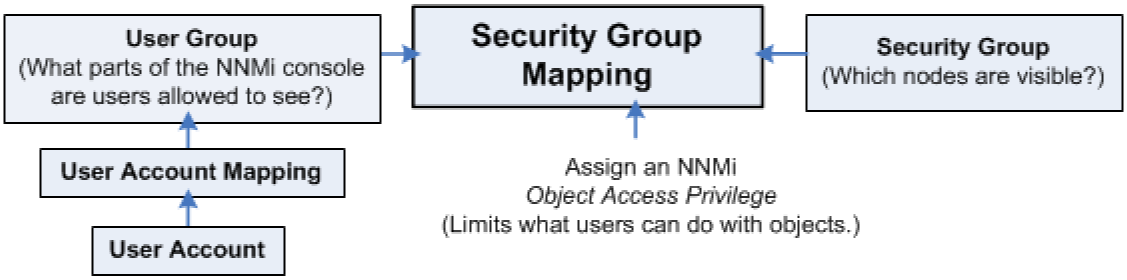
Configure Menu Items

The **Menu Items** tab of the **User Interface Configuration** option enables you to make changes or additions to the items available in the NNMi console menus. For example, you can configure Line Graphs (Graph Action) and additional NNMi actions (Launch Action) menu items that access in-house tools, Web sites, or a variety of other resources. See "Configure Menu Item Basic Details" on page 1417 for more information.

Chapter 13

Configuring Security

NNMi administrators configure security to meet the needs of their user environment.


User Group
(What parts of the NNMi console are users allowed to see?)
Security Group Mapping
Security Group
(Which nodes are visible?)
User Account Mapping
User Account
Assign an NNMi
Object Access Privilege
(Limits what users can do with objects.)

See "Determine Your Security Strategy" on page 507 for ideas.

> **Tip:** NNMi can be configured to use the Lightweight Directory Access Protocol (LDAP) and X.509 Certificates such as Public Key Infrastructure (PKI) user authentication for NNMi user names, passwords, and User Group Membership assignments. Additional steps are required. See "Choose a Mode for NNMi Access" on the next page.

NNMi enables an NNMi administrator to configure the following access control features:

Basis settings:	"About User Accounts" on page 513 "About User Groups" on page 513 "About User Account Mappings" on page 514
Required only for Operators and Guests: **Note:** NNMi administrators automatically see all nodes. NNMi users can have access to all nodes if they are a member of User Group: NNMi Global Operators.	"About Security Groups" on page 515 "About Security Group Mappings" on page 516

NNMi administrators can configure security in several ways:

"Using the Security Folder" on page 518

"Using the Security Wizard View" on page 524

nnmsecurity.ovpl command line tool

The NNMi administrator also needs to understand the following:

"Control Menu Access" on page 574

"Set Up Command Line Access to NNMi" on page 577

"Communicate Console Access Information to Your Team" on page 579

"About Multi-Tenancy and Global Network Management" on page 95

Verify that your NNMi Security configuration is working as expected:

"Troubleshoot NNMi Access" on page 582

Choose a Mode for NNMi Access

Decide how to configure access to NNMi:

- "NNMi Configuration Settings to Control NNMi Access" on the next page.

  NNMi user names, passwords, and User Group membership are defined within the NNMi database.

- "Lightweight Directory Access Protocol (LDAP) to Control NNMi Access" on page 506.

  NNMi administrators have choices about which information NNMi gathers from the directory service:

  a. User Accounts (user names and passwords)

  b. User Accounts (user names and passwords) plus User Groups and User Group Mappings

- "X.509 Certificates to Control NNMi Access" on page 507.

  The X.509 Certificate approach eliminates the need for any passwords.

  > **Tip:** NNMi supports Public Key Infrastructure (PKI) user authentication. This includes Smart Cards, such as Common Access Card (CAC) and Personal Identity Verification (PIV).

  NNMi administrators have choices about where NNMi gathers User Account Mapping information:

  a. NNMi's database

  b. Lightweight Directory Access Protocol (LDAP)

> **Caution:** You must choose *one* user authentication strategy and configure all NNMi users with the same approach.

User Authentication Strategy

Option	Which Method for User Authentication?	User Account Definitions in NNMi	User Group Definitions in NNMi	Which Method for Group Membership?
1 - Internal	NNMi Password	yes	yes	NNMi

User Authentication Strategy, continued

Option	Which Method for User Authentication?	User Account Definitions in NNMi	User Group Definitions in NNMi	Which Method for Group Membership?
				User Account Mappings
2 - Mixed	LDAP Password	yes	yes	NNMi User Account Mappings
	X.509 Certificate	yes	yes	NNMi User Account Mappings
3 - External	LDAP Password	no	yes	LDAP
	X.509 Certificate	no	yes	LDAP

* Assign each NNMi user to one or more User Groups. At a minimum, each NNMi user must belong to one of the following:

- NNMi Administrators
- NNMi Level 1 Operators
- NNMi Level 2 Operators
- Guests

NNMi Configuration Settings to Control NNMi Access

NNMi administrators configure NNMi user names, passwords, and NNMi User Group membership assignments in the NNMi database.

Which Database Stores the Information?

Mode	Using which User Authentication Method?	Are NNMi User Accounts Required?	Where is NNMi User Group Membership Assignment * defined?	Are NNMi User Groups & Mapping Required?
1	NNMi Password	Yes	NNMi	Yes

Caution: NNMi administrators must choose one Mode and configure all NNMi users with the same approach. See also:

- "Lightweight Directory Access Protocol (LDAP) to Control NNMi Access" on the next page
- "X.509 Certificates to Control NNMi Access" on page 507

To enable NNMi to store all user information in the NNMi database:

1. "Configure User Accounts (User Account Form)" on page 541.

   > **Tip:** NNMi administrators can also add, delete, or modify NNMi user names and passwords with the nnmsecurity.ovpl command-line tool.

2. "Configure User Groups (User Group Form)" on page 550.
3. "Map User Accounts to User Groups (User Account Mapping Form)" on page 553.

   NNMi users can belong to more than one User Group.

   The NNMi administrator must assign each User Account to a predefined NNMi User Group before that user can access NNMi. See "User Groups Provided in NNMi" on page 547 for more information.

4. "Configure Security Groups (Security Group Form)" on page 560
5. "Map User Groups to Security Groups (Security Group Mapping Form)" on page 565.

Lightweight Directory Access Protocol (LDAP) to Control NNMi Access

NNMi administrators can configure NNMi to rely on your environment's directory service to provide any of the following:

- Mixed: NNMi password
- External: NNMi password plus NNMi User Group membership assignments

User Authentication Strategy

Option	Which Method for User Authentication?	User Account Definitions in NNMi	User Group Definitions in NNMi	Which Method for Group Membership?
2 - Mixed	LDAP Password	yes	yes	NNMi User Account Mappings
3 - External	LDAP Password	no	yes	LDAP

> **Caution:** NNMi administrators must choose one Mode and configure all NNMi users with the same approach. See also:
>
> - "NNMi Configuration Settings to Control NNMi Access" on the previous page.
> - "X.509 Certificates to Control NNMi Access" on the next page.

Follow the instructions in the "Integrating NNMi with a Directory Service through LDAP" chapter in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `:http://h20230.www2.hp.com/selfsolve/manuals.`

X.509 Certificates to Control NNMi Access

The X.509 Certificate service eliminates the need for any NNMi passwords. NNMi administrators have a choice of where to define and store the required NNMi User Group membership assignments:

- Mixed: NNMi defines and stores the User Group assignments.
- External: NNMi uses the Lightweight Directory Access Protocol (LDAP) User Group assignments.

Tip: NNMi supports Public Key Infrastructure (PKI) user authentication. This includes Smart Cards, such as Common Access Card (CAC) and Personal Identity Verification (PIV).

User Authentication Strategy

Option	Which Method for User Authentication?	User Account Definitions in NNMi	User Group Definitions in NNMi	Which Method for Group Membership?
2 - Mixed	X.509 Certificate	yes	yes	NNMi User Account Mappings
3 - External	X.509 Certificate	no	yes	LDAP

Caution: NNMi administrators must choose one Mode and configure all NNMi users with the same approach. See also:

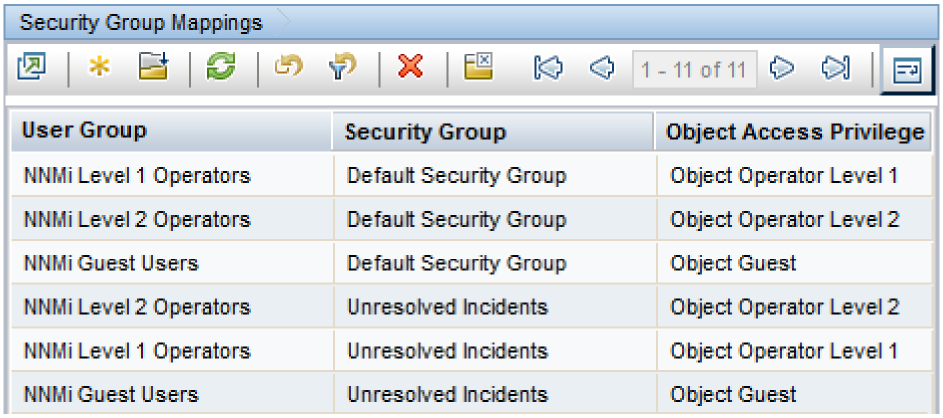
- "NNMi Configuration Settings to Control NNMi Access" on page 505
- "Lightweight Directory Access Protocol (LDAP) to Control NNMi Access" on the previous page.

Follow the instructions in the “Configuring NNMi to Support Public Key Infrastructure User Authentication” chapter in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.

Determine Your Security Strategy

Out-of-box, NNMi Security works in the following manner:

- NNMi assigns all nodes to the Default Security Group.
- NNMi operators and guests can see all discovered nodes and all incidents, because of the default Security Group Mappings:



User Group	Security Group	Object Access Privilege
NNMi Level 1 Operators	Default Security Group	Object Operator Level 1
NNMi Level 2 Operators	Default Security Group	Object Operator Level 2
NNMi Guest Users	Default Security Group	Object Guest
NNMi Level 2 Operators	Unresolved Incidents	Object Operator Level 2
NNMi Level 1 Operators	Unresolved Incidents	Object Operator Level 1
NNMi Guest Users	Unresolved Incidents	Object Guest

Tip: NNMi administrators always see all nodes and incidents, no Security Group Mappings are required for NNMi administrators.

NNMi administrators can limit access to nodes and incidents by deleting the default (out-of-box) Security Group Mappings. Then no operators or guests can access any nodes until an NNMi administrator explicitly adds new, more restrictive Security Group Mappings. When these out-of-box Security Group Mappings are removed, the predefined **NNMi User Group**[1]s provide access to the NNMi console only, rather than to the NNMi console and to all nodes. See "Remove User Groups from Security Group Mappings" on page 567 for more information.

Security Group Mappings have three components:

- User Group identifies the *NNMi users*.
- Security Group identifies *a set of nodes* (and indirectly their hosted objects).
- *Object Access Privilege* determines the level of access that each User Account in the User Group has to the nodes in the associated Security Group.

Each node is associated with one and only one Security Group. NNMi operators and guests can view a node only if one of the User Groups to which that NNMi user belongs is associated with that node's Security Group.

When NNMi discovers nodes in your network environment, Tenant and Security Group settings are established in the following manner:

- **Discovery Seeds**: If Nodes are discovered as Discovery seeds, the NNMi administrator specifies a Tenant for each Discovery Seed. See "Specify Discovery Seeds" on page 256. When NNMi administrators define a Tenant, they specify an **Initial Discovery Security Group**. Any newly discovered Node within the defined Tenant is assigned to this Security Group. NNMi administrators can change either the node's Tenant or Security Group assignment or both at any time.

  Nodes assigned to the *Default Security Group* are visible from all views. To control access to a device, assign that device to a Security Group other than Default Security Group.

[1] NNMi User Groups are those User Groups provided by NNMi. Users cannot access the NNMi console until their User Account is mapped to at least one of the following NNMi User Groups: NNMi Administrators, NNMi Level 2 Operators, NNMi Level 1 Operators (with less access privileges than Level 2 Operators), and NNMi Guest Users

Nodes within one Tenant can each be assigned to different Security Groups, and Nodes within one Security Group each be assigned to different Tenants.

- **Auto-Discovery for Default Tenant**: When you configure Auto-Discovery Rules, NNMi assigns any Nodes discovered using those Auto-Discovery Rules to the *Default Tenant* and whichever Security Group is currently configured as the Default Tenant's Initial Discovery Security Group setting (the *Default Security Group* out-of-box). See "Configure Tenants" on page 194 .

Global Network Management: Regional Managers forward information about Nodes to the Global Manager. The Global Manager's copy of the Node object has the same Tenant assignment as the Regional Manager's record of that Node.

In a Global Network Management environment, best practice is to have the NNMi administrators for the Global Manager and all Regional Managers agree to a predefined list of Tenant names. Those Tenants would be defined on the Regional Managers, the Tenant definitions exported, and those Tenant definitions imported onto the Global Manager (thus ensuring that the UUID and name value for each Tenant match on both NNMi management servers). The NNMi administrator on the Global Manager update their Tenant definitions to assign Initial Discovery Security Group values that make sense for the Global Manager's team. See "About Multi-Tenancy and Global Network Management" on page 95 for more information.

Note: If a Regional Manager forwards information about a Node to the Global Manager, and that Node is assigned to a Tenant object that does not exist on the Global Manager, NNMi creates a Tenant with the UUID and name from the Regional Manager, but creates a new Security Group with that Tenant name (does not duplicate the Regional Manager's setting for that Tenant's *Initial Discovery Security Group* setting). NNMi maps that new Security Group to the following:

- User Group = NNMi Administrator
- Object Access Privilege = Object Administrator

The Global Manager's NNMi administrator can assign a *different* Initial Discovery Security Group to a Tenant definition at any time. From that point onward, the NNMi Global Manager uses that new Initial Discovery Security Group setting when creating new nodes within that Tenant.

Node revisions: NNMi administrators can change the Node's initial Security Group assignment. See "Methods for Assigning Nodes to Security Groups" on page 563.

Tip: NNMi administrators can use Security Groups in Node Group definitions that become filters in NNMi views. If a user cannot access any nodes in a particular Node Group, that filter dynamically disappears from the filter selection list in the user's NNMi views. See "Specify Node Group Additional Filters" on page 298 for more information about Node Group filters.

Security influences incidents:

- Network operators and guests can view incidents associated with a node only if that user's User Account is mapped to one of the User Groups that are mapped to the node's Security Group. See "About Security Groups" on page 515 and "About Security Group Mappings" on page 516.
- Any incident that does not have an associated node is assigned to the **Unresolved Incidents** Security Group and NNMi's out-of-box configuration makes these incidents visible to all User

Groups. Examples of incidents that are unresolved include unresolved traps, system health, and license violation incidents.

- Operators should only be assigned incidents for nodes they can access.

The following examples present possible Security strategies. Consider printing one or more of the following topics to use as a tutorial about configuring NNMi Security. The Configure Security Tasks table explains all possible choices.

These strategy examples use the Security views under the Configuration workspace (see "Using the Security Folder" on page 518):

- "Configure Security: All Users Access All Nodes" on page 519
- "Configure Security: Limit Node Access" on page 520

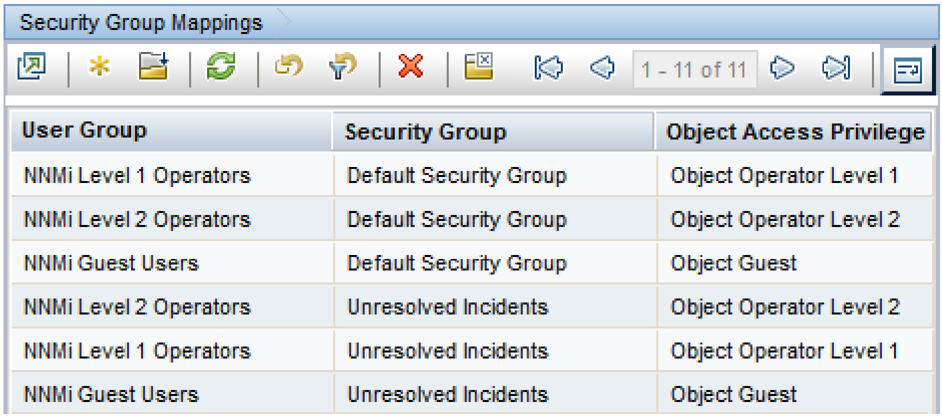
These strategy examples use the Security Wizard under the Configuration workspace (see "Using the Security Wizard View" on page 524):

- "Configure Security Example (Allow a Subset of Users to Access a Subset of Nodes)" on page 533
- "Configure Security Example (Divide Node Access Between Two or More User Groups)" on page 525

Configure Security Tasks

Task	Description
Determine your Security strategy.	Use the guidelines in this Help topic to understand how to configure Security for your network environment. You must also determine your users, their *Object Access Privileges*, and the nodes each user should access: "Control Menu Access" on page 574 "User Groups Provided in NNMi" on page 547 "Determine which NNMi User Group to Assign" on page 549
Remove the Default Security Group Mappings to NNMi User Groups	Out-of-box, NNMi assigns all Nodes to the Default Security Group and all NNMi users can see all Nodes. To ensure that none of your NNMi operators or guests can see nodes assigned to the **Default Security Group**, remove these out-of-box Security Group Mappings.

Configure Security Tasks, continued

<table>
<tr><th>Task</th><th>Description</th></tr>
<tr><td></td><td>


<table>
<tr><th>User Group</th><th>Security Group</th><th>Object Access Privilege</th></tr>
<tr><td>NNMi Level 1 Operators</td><td>Default Security Group</td><td>Object Operator Level 1</td></tr>
<tr><td>NNMi Level 2 Operators</td><td>Default Security Group</td><td>Object Operator Level 2</td></tr>
<tr><td>NNMi Guest Users</td><td>Default Security Group</td><td>Object Guest</td></tr>
<tr><td>NNMi Level 2 Operators</td><td>Unresolved Incidents</td><td>Object Operator Level 2</td></tr>
<tr><td>NNMi Level 1 Operators</td><td>Unresolved Incidents</td><td>Object Operator Level 1</td></tr>
<tr><td>NNMi Guest Users</td><td>Unresolved Incidents</td><td>Object Guest</td></tr>
</table>
<b>Note</b>: Deleting a Security Group Mapping does not delete the associated predefined NNMi User Group nor the <i>Object Access Privilege</i> definition.
</td></tr>
<tr><td>Configure User Accounts</td><td>You must create a User Account for each NNMi user.</td></tr>
<tr><td>Configure Additional User Groups</td><td>The NNMi administrator can create any number of User Groups to meet the needs of your network environment.
Examples of when additional User Groups are needed include the following circumstances:
• When you need a subset of users to access only a subset of nodes.
• When you need to divide node access between two or more User Groups (such as multiple shifts or multiple sites that share responsibilities).</td></tr>
<tr><td>Map User Accounts to the Predefined NNMi User Groups</td><td>A particular user cannot access the NNMi console until their User Account is mapped to at least one of the following predefined NNMi User Groups:
• NNMi Administrators
• NNMi Level 2 Operators
• NNMi Level 1 Operators (with less access privileges than Level 2 Operators)
• NNMi Guest Users
<b>Note:</b> NNMi provides two additional User Groups:
• NNMi Global Operators (<i>secondary</i>)
Assigning users to this <i>secondary</i> group, in addition to the user's currently assigned NNMi Guest User, NNMi Level 1 Operator, or NNMi Level 2 Operator assignment, provides access to all topology objects, but does not change any other aspect of their currently assigned NNMi Guest User, NNMi Level 1 Operator, or NNMi Level 2 Operator assignment.</td></tr>
</table>

Configure Security Tasks, continued

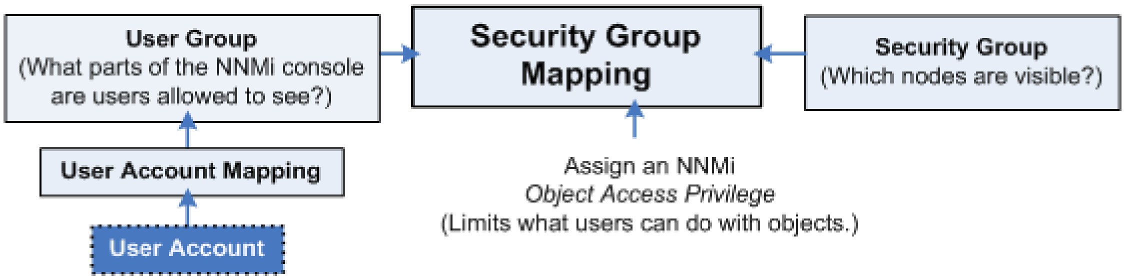
Task	Description
	Users assigned to the NNMi Administrators User Group do not need any *secondary* group assignment. These users already can access all topology objects. • NNMi Web Services Client Used *only to provide access for software* that is integrated with NNMi. See "Integrations with HP and Third-Party Products" on page 1487 - for example, "HP RAMS MPLS WAN Configuration (NNMi Advanced)" on page 1410). Do not use any other User Group for software integrations.
Map User Accounts to Additional User Groups	If you created additional User Groups, map the appropriate User Accounts to each User Group you created.
Configure Security Groups	By default, all operators can access all nodes discovered by NNMi. However, the NNMi administrator can limit visibility to a subset of nodes for some or all operators by using User Groups and Security Groups. **Note**: Each node can be mapped to one and only one Security Group. Examples of when you need to create additional Security Groups to limit node access include the following circumstances: • When you need a subset of users to access only a subset of nodes. • When you need to divide node access between two or more User Groups
Map Security Groups to User Groups	After creating any additional User Groups, you map each User Group to a Security Group and assign the *Object Access Privilege* for this Security Group Mapping. The *Object Access Privilege* determines the level of access that each User Group has to the nodes that are visible. Users can view a node only if one of the User Groups to which they belong is associated with that node's Security Group.
Assign Nodes to Security Groups	Out-of-box, NNMi Security settings allow all NNMi User Groups to access nodes assigned to the Default Security Group. If you create Security Groups to limit node access, you must assign nodes to the appropriate Security Group. Each node is associated with one and only one Security Group.
Verify Your Configuratio n Changes	NNMi provides a report that includes information about any of the following potential problems: • Users Accounts that are not mapped to a User Group • User Accounts that are not mapped to an NNMi User Group

Configure Security Tasks, continued

Task	Description
	• User Accounts that have unusual NNMi role combinations • Security Groups that include nodes from multiple tenants • Empty User Groups and Security Groups • Tenants with the same name • Security Groups with the same name

About User Accounts

The NNMi administrator configures User Accounts as part of the Security Configuration that controls who accesses the NNMi console.


User Group
(What parts of the NNMi console are users allowed to see?)
Security Group Mapping
Security Group
(Which nodes are visible?)
User Account Mapping
Assign an NNMi
Object Access Privilege
(Limits what users can do with objects.)
User Account

Each User Account represents a user.

NNMi administrators can configure User Accounts using the following methods:

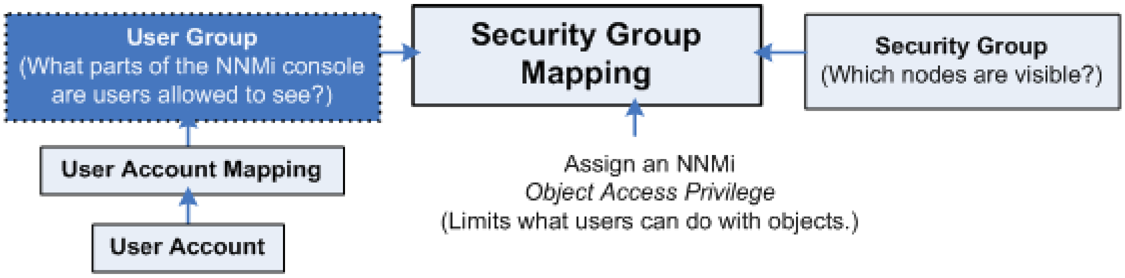
- The Configuration Wizard ("Create and Delete User Accounts Using the Security Wizard" on page 545)
- The User Accounts view ("Configure User Accounts (User Account Form)" on page 541)
- The nnmsecurity.ovpl command line tool

NNMi can be configured to use the Lightweight Directory Access Protocol (LDAP) and X.509 Certificates such as Public Key Infrastructure (PKI) user authentication for NNMi user names, passwords, and User Group Membership assignments. Additional steps are required. See "Choose a Mode for NNMi Access" on page 504.

Next step: "About User Groups" below

About User Groups

The NNMi administrator configures User Groups as part of the Security Configuration that controls who accesses the NNMi console.


User Group
(What parts of the NNMi console are users allowed to see?)
Security Group Mapping
Security Group
(Which nodes are visible?)
User Account Mapping
User Account
Assign an NNMi
Object Access Privilege
(Limits what users can do with objects.)

NNMi provides the following predefined User Groups (NNMi users cannot access the NNMi console until their User Account is mapped to at least one of these). The predefined NNMi User Group that the NNMi administrator assigns to each User Account determines which workspaces, views, menus, actions, and object attributes are visible to each user within the NNMi console (see "User Groups Provided in NNMi" on page 547 for details):

- NNMi Administrators (no Security Group Mapping required)
- NNMi Level 2 Operators
- NNMi Level 1 Operators (with less access privileges than Level 2 Operators)
- NNMi Guest Users

NNMi administrators can configure User Accounts using the following methods:

- The Configuration Wizard ("Create and Delete User Groups Using the Security Wizard" on page 552)
- The User Accounts view ("Configure User Groups (User Group Form)" on page 550)
- The nnmsecurity.ovpl command line tool

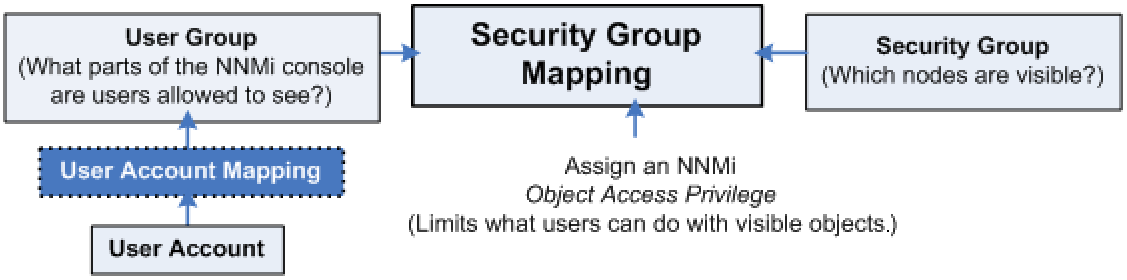
NNMi administrators can also create additional User Groups to fine tune NNMi access. See "Determine Your Security Strategy" on page 507.

NNMi can be configured to use the Lightweight Directory Access Protocol (LDAP) and X.509 Certificates such as Public Key Infrastructure (PKI) user authentication for NNMi user names, passwords, and User Group Membership assignments. Additional steps are required. See "Choose a Mode for NNMi Access" on page 504.

Next step: "About User Account Mappings" below

About User Account Mappings

User Account Mappings enable the NNMi administrator to assign a User Account to one or more User Groups to control NNMi console access.


User Group
(What parts of the NNMi console are users allowed to see?)
Security Group Mapping
Security Group
(Which nodes are visible?)
User Account Mapping
User Account
Assign an NNMi
Object Access Privilege
(Limits what users can do with visible objects.)

At least one predefined NNMi User Group must be mapped to each User Account to determine which workspaces, views, menus, actions, and object attributes are visible to that User Account within the NNMi console. See "About User Accounts" on page 513 and "About User Groups" on page 513 and "User Groups Provided in NNMi" on page 547 for details.

A User Account can be mapped to two or more User Groups. NNMi administrators can create any number of User Groups.

A User Account Mapping is a separate object in the NNMi database. Therefore, when you create or delete a User Account Mapping, you create or delete only the User Account Mapping, not the User Account or User Group.

NNMi administrators can map User Accounts to User Groups using the following methods:

- The Configuration Wizard ("Map User Accounts and User Groups " on page 556)
- The User Account Mappings view ("Map User Accounts to User Groups (User Account Mapping Form)" on page 553)
- The nnmsecurity.ovpl command line tool

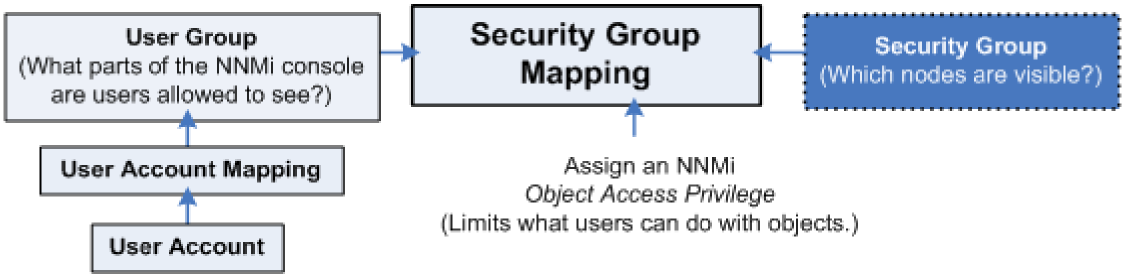
NNMi can be configured to use the Lightweight Directory Access Protocol (LDAP) and X.509 Certificates such as Public Key Infrastructure (PKI) user authentication for NNMi user names, passwords, and User Group Membership assignments. Additional steps are required. See "Choose a Mode for NNMi Access" on page 504.

Next step: "About Security Groups" below (only for Operator or Guest users)

About Security Groups

Required only for Operator or Guest users:

The NNMi administrator configures Security Groups as part of the Security Configuration that controls which nodes are accessed in the NNMi console. (NNMi administrators automatically see all nodes.)


User Group
(What parts of the NNMi console are users allowed to see?)
Security Group Mapping
Security Group
(Which nodes are visible?)
User Account Mapping
User Account
Assign an NNMi
Object Access Privilege
(Limits what users can do with objects.)

Security Groups define sets of nodes within your network environment. Each node is assigned to only one Security Group. Your security strategy determines the number of Security Groups required for your network environment. See "Determine Your Security Strategy" on page 507. Out-of-box, NNMi assigns all nodes to the **Default Security Group** and all NNMi users see those nodes (based on the out-of-box Security Group Mappings).

NNMi administrators can configure Security Groups to limit node access by using the following methods:

- The Configuration Wizard ("Create and Delete Security Groups Using the Security Wizard" on page 561)
- The Security Accounts view ("Configure Security Groups (Security Group Form)" on page 560)
- The nnmsecurity.ovpl command line tool

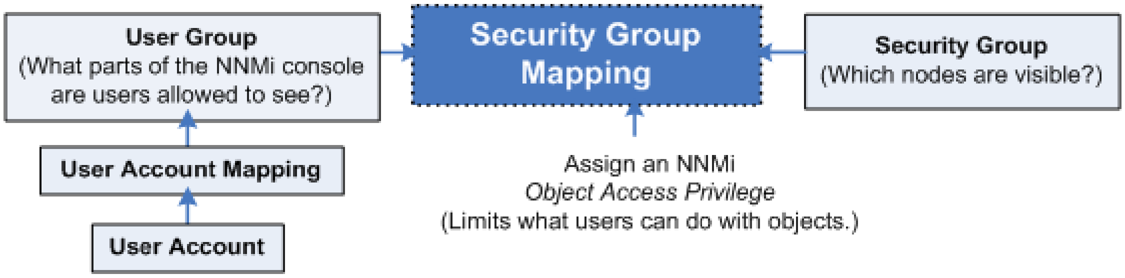
The NNMi administrator can assign Nodes to Security Groups. See "Methods for Assigning Nodes to Security Groups" on page 563.

Next step: "About Security Group Mappings" below (only for Operator or Guest users)

About Security Group Mappings

Required only for Operator or Guest users:

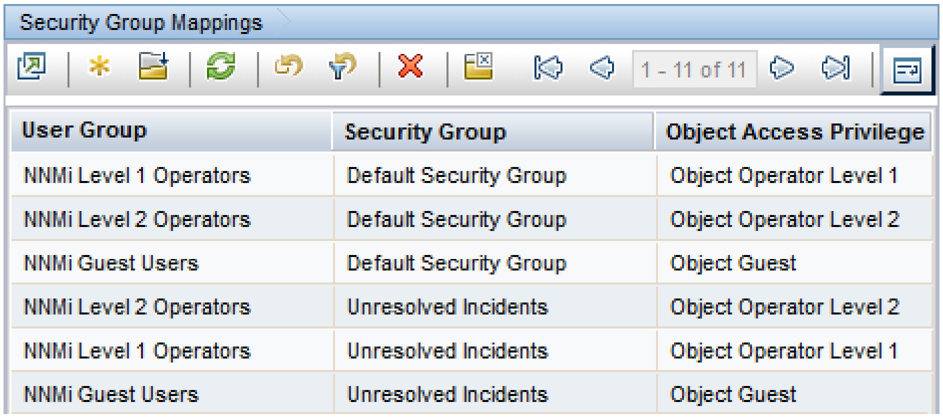
Security Group Mappings control which nodes are visible to NNMi operators and guests, and what NNMi operators and guests can do with those visible nodes. (Security Group Mappings are irrelevant to users assigned to the *NNMi Administrators* User Group. NNMi administrators automatically see all nodes and have full access rights.)


User Group
(What parts of the NNMi console are users allowed to see?)
Security Group Mapping
Security Group
(Which nodes are visible?)
User Account Mapping
User Account
Assign an NNMi
Object Access Privilege
(Limits what users can do with objects.)

Security Group Mappings have three components:

1. "About User Groups" on page 513
2. "About Security Groups" on the previous page
3. "Object Access Privileges Provided in NNMi" on page 566

NNMi provides the following *default* Security Group Mappings that allow all NNMi operators and guests to see all Nodes and all incidents that are not associated with any particular node. NNMi administrators can delete these *default* mappings and create new mappings that provide more limited control. (Deleting a Security Group Mapping does not delete the associated User Group or Security Group, so NNMi administrators can then map those User Groups and Security Groups in other ways with more limited control.)



User Group	Security Group	Object Access Privilege
NNMi Level 1 Operators	Default Security Group	Object Operator Level 1
NNMi Level 2 Operators	Default Security Group	Object Operator Level 2
NNMi Guest Users	Default Security Group	Object Guest
NNMi Level 2 Operators	Unresolved Incidents	Object Operator Level 2
NNMi Level 1 Operators	Unresolved Incidents	Object Operator Level 1
NNMi Guest Users	Unresolved Incidents	Object Guest

NNMi provides predefined *Object Access Privileges*. The Object Access Privilege determines the level of access that each User Group has to the visible nodes. Level of node access includes the actions that can be performed on the nodes. See "Object Access Privileges Provided in NNMi" on page 566.

For example, if an NNMi operator is mapped to a User Group with **NNMi Level 2 Operators**, but their Security Group Mapping's *Object Access Privilege* is **Object Operator Level 1** (with less access privileges than Level 2), that NNMi operator *sees* all of the actions available to NNMi Level 2 Operators, but can run only those *actions allowed* for NNMi Level 1 Operators.

If an NNMi operator or guest is assigned to multiple Security Group Mappings

- Multiple predefined **NNMi User Group**[1]s, the NNMi console displays all the parts of NNMi that are available to the highest User Group.
- Multiple *Object Access Privileges*, actions available for each node are determined by the node's Security Group Mapping. If mapped to the same Security Group multiple times, the highest access level is available.

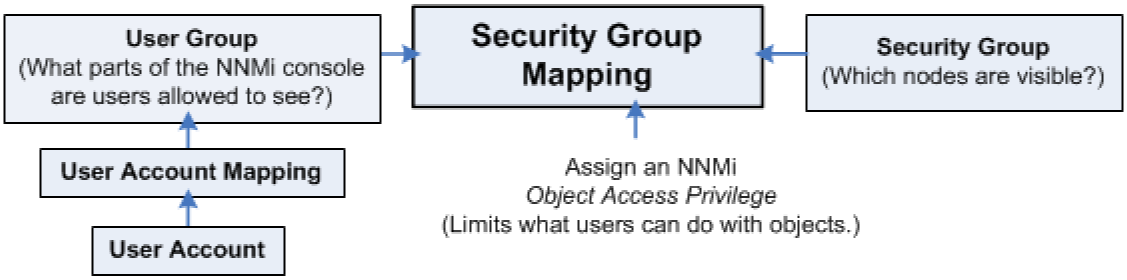
NNMi administrators can map User Groups to Security Groups using the following methods:

- The Configuration Wizard ("Map User Groups and Security Groups " on page 569)
- The Security Accounts view ("Map User Groups to Security Groups (Security Group Mapping Form)" on page 565)
- The nnmsecurity.ovpl command line tool

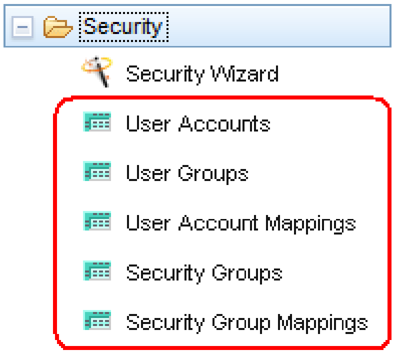
Next step: "Check Security Configuration" on page 584

[1] NNMi User Groups are those User Groups provided by NNMi. Users cannot access the NNMi console until their User Account is mapped to at least one of the following NNMi User Groups: NNMi Administrators, NNMi Level 2 Operators, NNMi Level 1 Operators (with less access privileges than Level 2 Operators), and NNMi Guest Users

Using the Security Folder


User Group
(What parts of the NNMi console are users allowed to see?)
Security Group Mapping
Security Group
(Which nodes are visible?)
User Account Mapping
User Account
Assign an NNMi
Object Access Privilege
(Limits what users can do with objects.)

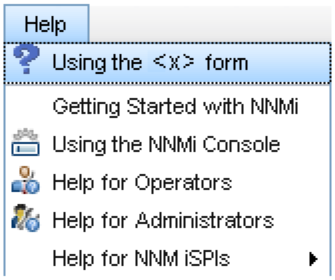
NNMi enables an NNMi administrator to configure the following configurations using Security workspace views:



Tip: Select **Help** → **System Information** to view the User Name, NNMi Role, and User Group for the current NNMi session.

To configure Security using the Security workspace:

1. Determine your Security strategy (see "Determine Your Security Strategy" on page 507).
2. Navigate to the **Security** workspace.
3. Make your configuration choices using the Security views. Refer to the About the <x> form Help available for each form within the Security views.

   NNMi's security model restricts access to the NNMi console based on User Account to User Group mappings. An NNMi administrator can also choose to restrict Node access based on Security Groups and Security Group Mappings (User Group to Security Group).

   Two examples are provided. Use these examples as a guideline for configuring security.

- "Configure Security: All Users Access All Nodes" below
- "Configure Security: Limit Node Access" on the next page

Note: You can also configure security using the Security Folder in the Configuration workspace. See "Using the Security Wizard View" on page 524 for more information.

4. Click **Save and Close**.
5. See "Methods for Assigning Nodes to Security Groups" on page 563.

Configure Security: All Users Access All Nodes

If you want all of your NNMi users to access all of the nodes discovered by NNMi, use these guidelines.

Note: You can also use the nnmsecurity.ovpl command to configure User Accounts, User Groups, Security Groups, and Tenants.

> **Tip:** Select **Help** → **System Information** to view the User Name, NNMi Role, and User Group for the current NNMi session.

To configure Security:

1. Navigate to the **Security** workspace.
2. Make your configuration choices (see table).
3. Click **Save and Close**.

Configure Security Tasks (Using the Security workspace)

Task	Description
Determine your users and their **NNMi User Group**[1] or Groups	See "Determine Your Security Strategy" on page 507 and the following topics: "Control Menu Access" on page 574 "User Groups Provided in NNMi" on page 547 "Determine which NNMi User Group to Assign" on page 549
Configure User Accounts	You must create a User Account for each NNMi user.
Map User Accounts to the Predefined	A particular user cannot access the NNMi console until their User Account is mapped to at least one of the following predefined default NNMi User Groups: • NNMi Administrators

[1] NNMi User Groups are those User Groups provided by NNMi. Users cannot access the NNMi console until their User Account is mapped to at least one of the following NNMi User Groups: NNMi Administrators, NNMi Level 2 Operators, NNMi Level 1 Operators (with less access privileges than Level 2 Operators), and NNMi Guest Users

Configure Security Tasks (Using the Security workspace), continued

Task	Description
NNMi User Groups	• NNMi Level 2 Operators • NNMi Level 1 Operators (with less access privileges than Level 2 Operators) • NNMi Guest Users **Note:** NNMi provides two additional User Groups: • NNMi Global Operators (*secondary*) Assigning users to this *secondary* group, in addition to the user's currently assigned NNMi Guest User, NNMi Level 1 Operator, or NNMi Level 2 Operator assignment, provides access to all topology objects, but does not change any other aspect of their currently assigned NNMi Guest User, NNMi Level 1 Operator, or NNMi Level 2 Operator assignment. Users assigned to the NNMi Administrators User Group do not need any *secondary* group assignment. These users already can access all topology objects. • NNMi Web Services Client Used *only to provide access for software* that is integrated with NNMi. See "Integrations with HP and Third-Party Products" on page 1487 - for example, "HP RAMS MPLS WAN Configuration (NNMi Advanced)" on page 1410). Do not use any other User Group for software integrations.
Verify Your Configuration Changes	NNMi provides a report that includes information about any of the following potential problems: • Users Accounts that are not mapped to a User Group • User Accounts that are not mapped to an NNMi User Group • User Accounts that have unusual NNMi role combinations • Security Groups that include nodes from multiple tenants • Empty User Groups and Security Groups • Tenants with the same name • Security Groups with the same name

Configure Security: Limit Node Access

To limit node access, use these guidelines. Ways you might limit node access include the following:

- To permit a subset of users to access only a subset of nodes.
- To divide node access between two or more User Groups

> **Note:** You can also use the nnmsecurity.ovpl command to configure User Accounts, User Groups, Security Groups, and Tenants.

> **Tip:** Select **Help** → **System Information** to view the User Name, NNMi Role, and User Group for the current NNMi session.

To configure Security:

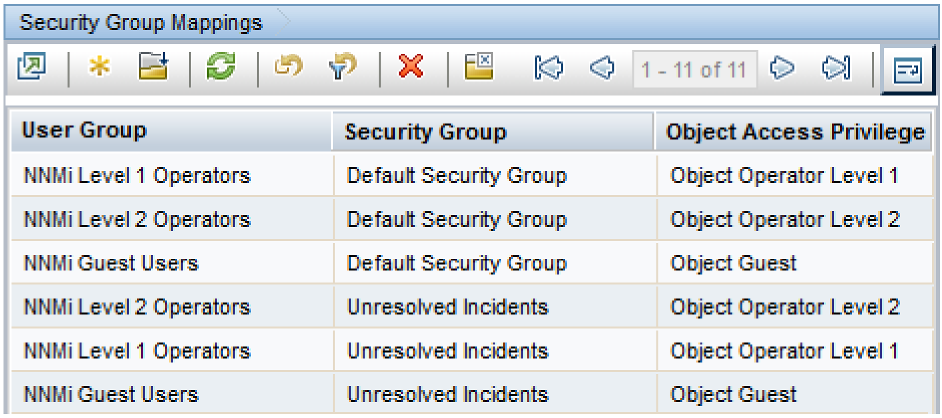
1. Navigate to the **Security** workspace.
2. Make your configuration choices (see table).
3. Click **Save and Close**.

Also see:

"Configure Security Example (Allow a Subset of Users to Access a Subset of Nodes)" on page 533

"Configure Security Example (Divide Node Access Between Two or More User Groups)" on page 525

Configure Security Tasks (Limit Node Access)

<table>
<tr><th>Task</th><th>Description</th></tr>
<tr><td>Determine your users, their privileges, and the nodes that each user each should access.</td><td>See "Determine Your Security Strategy" on page 507 and the following topics:
"Control Menu Access" on page 574
"User Groups Provided in NNMi" on page 547
"Determine which NNMi User Group to Assign" on page 549</td></tr>
<tr><td>Remove the Default Security Group Mapping to NNMi User Groups</td><td>To ensure that none of your NNMi operators or guests can see nodes assigned to the <b>Default Security Group</b>, remove the out-of box Security mappings.

Security Group Mappings
1 - 11 of 11
<table>
<tr><th>User Group</th><th>Security Group</th><th>Object Access Privilege</th></tr>
<tr><td>NNMi Level 1 Operators</td><td>Default Security Group</td><td>Object Operator Level 1</td></tr>
<tr><td>NNMi Level 2 Operators</td><td>Default Security Group</td><td>Object Operator Level 2</td></tr>
<tr><td>NNMi Guest Users</td><td>Default Security Group</td><td>Object Guest</td></tr>
<tr><td>NNMi Level 2 Operators</td><td>Unresolved Incidents</td><td>Object Operator Level 2</td></tr>
<tr><td>NNMi Level 1 Operators</td><td>Unresolved Incidents</td><td>Object Operator Level 1</td></tr>
<tr><td>NNMi Guest Users</td><td>Unresolved Incidents</td><td>Object Guest</td></tr>
</table></td></tr>
</table>

Configure Security Tasks (Limit Node Access), continued

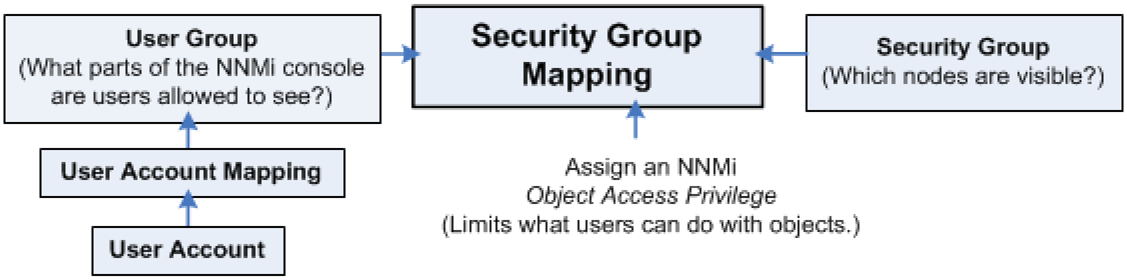
Task	Description
	Note: Deleting a Security Group Mapping does not delete the associated predefined NNMi User Group nor the *Object Access Privilege* definition.
Configure User Accounts	You must create a User Account for each NNMi user.
Configure Additional User Groups	Out-of-box, all operators and guests can access all nodes discovered by NNMi. However, the NNMi administrator can limit visibility to parts of the network for operators and guests with User Groups and Security Groups. Examples of when additional User Groups are needed include the following circumstances: • To permit a subset of users to access only a subset of nodes • To divide node access between two or more User Groups
Map User Accounts to the Predefined NNMi User Groups	A particular user cannot access the NNMi console until their User Account is mapped to at least one predefined **NNMi User Group**[1]: • NNMi Administrators • NNMi Level 2 Operators • NNMi Level 1 Operators (with less access privileges than Level 2 Operators) • NNMi Guest Users **Note:** NNMi provides two additional User Groups: • NNMi Global Operators (*secondary*) Assigning users to this *secondary* group, in addition to the user's currently assigned NNMi Guest User, NNMi Level 1 Operator, or NNMi Level 2 Operator assignment, provides access to all topology objects, but does not change any other aspect of their currently assigned NNMi Guest User, NNMi Level 1 Operator, or NNMi Level 2 Operator assignment. Users assigned to the NNMi Administrators User Group do not need any *secondary* group assignment. These users already can access all topology objects. • NNMi Web Services Client Used *only to provide access for software* that is integrated with NNMi. See "Integrations with HP and Third-Party Products" on page 1487 - for

[1] NNMi User Groups are those User Groups provided by NNMi. Users cannot access the NNMi console until their User Account is mapped to at least one of the following NNMi User Groups: NNMi Administrators, NNMi Level 2 Operators, NNMi Level 1 Operators (with less access privileges than Level 2 Operators), and NNMi Guest Users

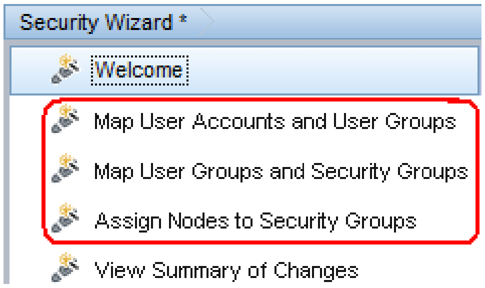
Configure Security Tasks (Limit Node Access), continued

Task	Description
	example, "HP RAMS MPLS WAN Configuration (NNMi Advanced)" on page 1410). Do not use any other User Group for software integrations.
Map User Accounts to Additional User Groups	Map the appropriate User Accounts to each User Group that you created.
Configure Security Groups	Configure a Security Group for each set of nodes that requires limited access. **Note:** Each node can be mapped to one and only one Security Group. For example, if you want to limit access to nodes in a single location, such as Los Angeles, create a Los Angeles Security Group.
Assign Nodes to Security Groups	If you create Security Groups to limit node access, you must assign nodes to the appropriate Security Group. **Note:** Each node can be mapped to one and only one Security Group.
Map Security Groups to User Groups	Users can view a node only if one of the User Groups to which they belong is associated with that node's Security Group. Map each User Group to one or more Security Groups. **Note:** When NNMi administrators map a User Group to a Security Group, they assign the **Object Access Privilege** for this Security Group Mapping. The *Object Access Privilege* determines the level of access that each User Group has to the nodes that are visible to it.
Verify Your Configuration Changes	NNMi provides a report that includes information about any of the following potential problems: • User Accounts that are not mapped to a User Group • User Accounts that are not mapped to an NNMi User Group • User Accounts that have unusual NNMi role combinations • Security Groups that include nodes from multiple tenants • Empty User Groups and Security Groups • Tenants with the same name • Security Groups with the same name

Using the Security Wizard View


User Group
(What parts of the NNMi console are users allowed to see?)
Security Group Mapping
Security Group
(Which nodes are visible?)
User Account Mapping
User Account
Assign an NNMi
Object Access Privilege
(Limits what users can do with objects.)

These Configuring Security Wizard pages enables NNMi administrators to configure the following access control features. You can access the wizard pages in any order:


Security Wizard *
Welcome
Map User Accounts and User Groups
Map User Groups and Security Groups
Assign Nodes to Security Groups
View Summary of Changes

- On the Map User Accounts and User Groups page:
  - User Accounts
  - User Groups
  - User Account / Group Mappings
- On the Assign Nodes to Security Groups page:

  Security Groups
- On the Map User Groups and Security Groups:

  Security Group Mappings

To configure Security using the Security wizard:

1. Determine your Security strategy (see table).
2. Navigate to the **Security Wizard**.
   a. From the Workspaces navigation panel, select the **Configuration** workspace.
   b. Expand **Security**.
   c. Select **Security Wizard**.
3. Make your configuration choices. Refer to the links to online Help from within the Discovery Wizard.

NNMi's security model restricts access to the NNMi console based on User Account to User Group mappings. An NNMi administrator can also choose to restrict Node access based on Security Groups and Security Group Mappings (User Group to Security Group).

Two examples of using the Security Wizard are provided.

Tip: Use these examples as a guideline for configuring security.

Select the example that best matches your security configuration requirements:

- "Configure Security Example (Allow a Subset of Users to Access a Subset of Nodes)" on page 533
- "Configure Security Example (Divide Node Access Between Two or More User Groups)" below

Note: You can also configure security using the Security Folder in the Configuration workspace. See "Using the Security Folder" on page 518 for more information.

4. Click **Save and Close**.
5. See "Methods for Assigning Nodes to Security Groups" on page 563.

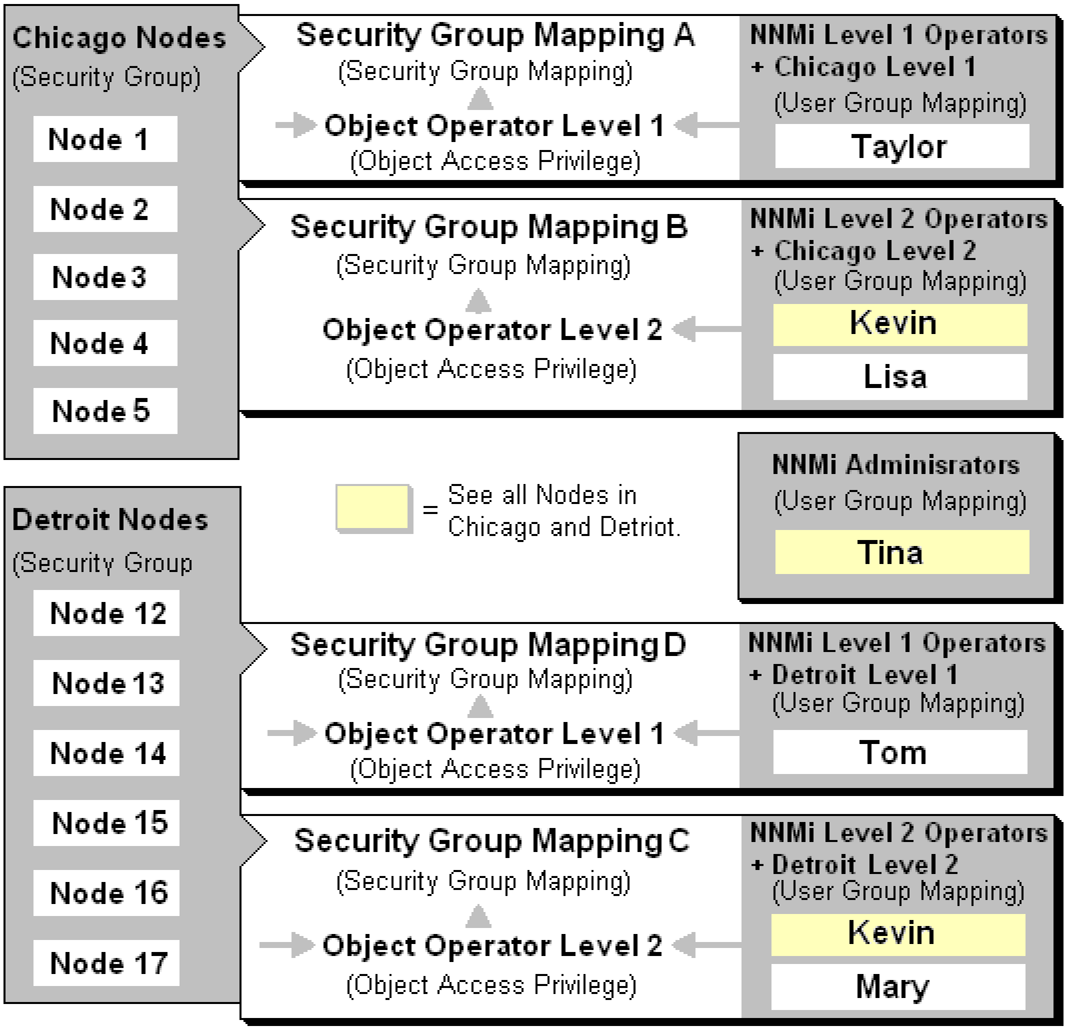
Configure Security Example (Divide Node Access Between Two or More User Groups)

This example uses NNMi's security configuration to divide the responsibility for network monitoring based on the following locations:

- Chicago
- Detroit

Each location includes an NNMi Level 1 Operator (with less access privileges than Level 2 Operators) and an NNMi Level 2 Operator. Tina, the NNMi Administrator, handles both locations. Kevin is a backup for both Chicago and Detroit and must access the nodes in both Chicago and Detroit.

The following diagram illustrates the security requirements:



The following table lists the NNMi console (**NNMi User Group**[1]) and node access requirements (User Group, Object Access Privilege and Security Group) for each location.

> **Note:** You can place all operators into the NNMi Level 2 Operators if you want all operators to see all menu options, but only have the ability to run them based on their Object Access Privilege.

Example Security Configuration

User Accounts	NNMi User Groups	User Groups	Object Access Privileges	Security Groups
Tina	NNMi Administrator	Not Applicable. The NNMi	Not Applicable. The NNMi Administrator has	Not Applicable. The NNMi

[1] NNMi User Groups are those User Groups provided by NNMi. Users cannot access the NNMi console until their User Account is mapped to at least one of the following NNMi User Groups: NNMi Administrators, NNMi Level 2 Operators, NNMi Level 1 Operators (with less access privileges than Level 2 Operators), and NNMi Guest Users

Example Security Configuration, continued

User Accounts	NNMi User Groups	User Groups	Object Access Privileges	Security Groups
		Administrator can access all nodes.	Administrator privileges to all nodes.	Administrator can access all nodes.
Kevin	NNMi Level 2 Operators	Chicago Level 2 Detroit Level 2	Object Operator Level 2	Chicago Nodes, Detroit Nodes
Lisa	NNMi Level 2 Operators	Chicago Level 2	Object Operator Level 2	Chicago Nodes
Taylor	NNMi Level 1 Operators	Chicago Level 1	Object Operator Level 1	Chicago Nodes
Mary	NNMi Level 2 Operators	Detroit Level 2	Object Operator Level 2	Detroit Nodes
Tom	NNMi Level 1 Operators	Detroit Level 1	Object Operator Level 1	Detroit Nodes

To set up security for the Chicago and Detroit locations follow these procedures:

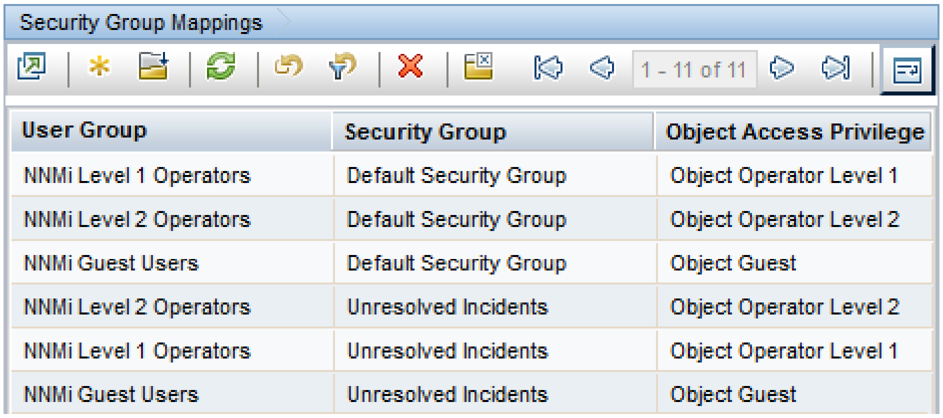
- Remove the Default Security Group Mapping to NNMi User Groups: NNMi Level 1 Operators, NNMi Level 2 Operators, and NNMi Guest

  > **Note:** The NNMi User Groups are provided for those NNMi administrators who are not concerned with Security configuration. After you remove these Security Group Mappings, the NNMi User Groups provide access to the NNMi console only rather than to the NNMi console and to all nodes.

- Create the User Accounts. (See the Example Security Configuration table.)
- Create the Additional User Groups required for the Chicago and Detroit Security Groups (Chicago Level 2, Chicago Level 1, Detroit Level 2, Deteroit Level 1). (See the Example Security Configuration table.)
- Map User Accounts to NNMi User Groups. (See the Example Security Configuration table.)
- Create the Security Groups for each location.
- Map each Security Group to the new User Groups. (See the Example Security Configuration table.)
- Assign the nodes to the appropriate Security Group.
- View a summary of your configuration changes

Remove the Default Security Group Mapping to NNMi User Groups

1. From the **Security Wizard** main page, select the **Map User Groups and Security Groups** option.
2. Navigate to the **Security Group Mappings** table.



User Group	Security Group	Object Access Privilege
NNMi Level 1 Operators	Default Security Group	Object Operator Level 1
NNMi Level 2 Operators	Default Security Group	Object Operator Level 2
NNMi Guest Users	Default Security Group	Object Guest
NNMi Level 2 Operators	Unresolved Incidents	Object Operator Level 2
NNMi Level 1 Operators	Unresolved Incidents	Object Operator Level 1
NNMi Guest Users	Unresolved Incidents	Object Guest

3. Click the row representing the **NNMi Level 1 Operators** User Group.

4. Click the ✖ Delete icon to remove the Default Security Group to NNMi Level 1 Operators User Group mapping.

5. Repeat steps 3 and 4 to remove the Default Security Group to **NNMi Level 2 Operator** and the **NNMi Guest** User Group mappings.

6. Continue or, click the **Save and Close** button to save your security configuration:

> **Note:** NNMi does not save any configuration changes until after you click **Save and Close** to save your security configuration.

Create User Accounts

1. In the Configuration workspace, select **Security Wizard**

2. From the **Security Wizard** main page, select the **Map User Accounts and User Groups** option.

3. Navigate to the **User Accounts** table.

4. Click * **New**.

5. In the **Create User Account** dialog box, enter the following:

   a. **Name**: Enter the user name **Tina**.

   b. **Password**: Enter the Password value **Tina**. The Password value can be up to 40 alphanumeric characters, punctuation, spaces, and underline characters.

   c. **Directory Service Account**: Do not enable this option.

      □ = User name and password are stored in the NNMi database. See "NNMi Configuration Settings to Control NNMi Access" on page 505.

6. Click **Add**.

7. Repeat steps 5 and 6 to add each User Account. (See the Example Security Configuration table.)

8. When you finish creating User Accounts, in the **Create User Account** dialog box, click **Close**

.

9. Continue or, click the **Save and Close** button to save your security configuration:

   > **Note:** NNMi does not save any configuration changes until you click **Save and Close** to save your security configuration.

Create Additional User Groups

1. From the **Security Wizard** main page, select the **Map User Accounts and User Groups** option
2. Navigate to the **User Groups** table.
3. Click ✱ **New**.
4. In the **Create User Group** dialog box, enter the following:
   a. **Name**: Enter **ChicagoLevel2**. The name can be a maximum of 40 alpha-numeric characters. Spaces are not permitted.

   b. **Display Name**: Enter **Chicago Level 2**. The Display Name is displayed in the NNMi console to identify this User Group. Enter a maximum of 50 characters. Alpha-numeric, spaces, and special characters (~ ! @ # $ % ^ &amp; * ( ) _+ -) are permitted.

   c. **Directory Service Name**: *Optional*. When Lightweight Directory Access Protocol (LDAP) define this User Group, enter the group's Distinguished Name. See the following topics:

      - "Lightweight Directory Access Protocol (LDAP) to Control NNMi Access" on page 506.
      - "X.509 Certificates to Control NNMi Access" on page 507

   d. **Description**: Type a maximum of 2048 characters to describe this User Group. Alpha-numeric, spaces, and special characters (~ ! @ # $ % ^ &amp; * ( ) _+ -) are permitted.
5. Click **Add**.
6. Repeat steps 2 and 3 to add each User Group. (See the Example Security Configuration table.)
7. When you finish creating User Groups, in the **Create User Group** dialog box, click **Close**

.

8. Continue or, click the **Save and Close** button to save your security configuration:

   > **Note:** NNMi does not save any configuration changes until you click **Save and Close** to save your security configuration.

Map User Accounts to User Groups

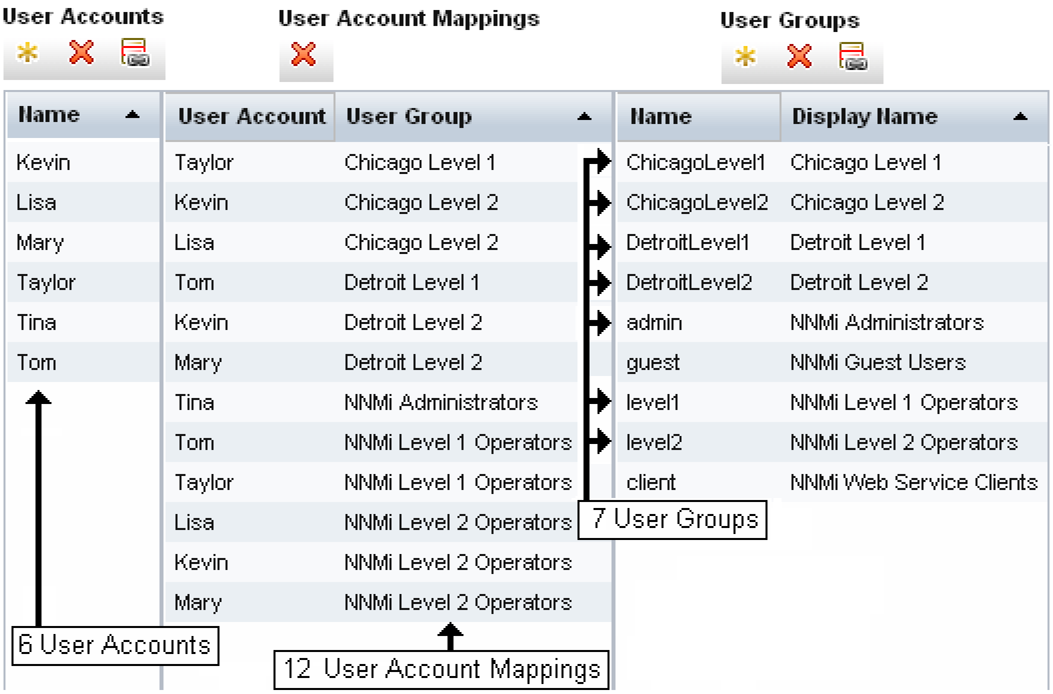
> **Note:** A User Account cannot access the NNMi console until it is mapped to one of the NNMi User Groups.

1. From the **Security Wizard** main page, select the **Map User Accounts and User Groups** option
2. Select Tina in the **User Accounts** table.
3. In the **User Groups** table, select the left arrow that precedes the **NNMi Administrators** User Group.

   The User Account and User Group names appear in the **User Account Mapping** table.
4. Repeat steps 1 and 2 to assign each User Account to the appropriate User Group. (See the Example Security Configuration table.)
5. Continue or, click the **Save and Close** button to save your security configuration:

   > **Note:** NNMi does not save any configuration changes until you click **Save and Close** to save your security configuration.

   Your User Account to User Group mappings should look similar to the following:



User Accounts

Name
Kevin
Lisa
Mary
Taylor
Tina
Tom

6 User Accounts

User Account Mappings

User Account	User Group
Taylor	Chicago Level 1
Kevin	Chicago Level 2
Lisa	Chicago Level 2
Tom	Detroit Level 1
Kevin	Detroit Level 2
Mary	Detroit Level 2
Tina	NNMi Administrators
Tom	NNMi Level 1 Operators
Taylor	NNMi Level 1 Operators
Lisa	NNMi Level 2 Operators
Kevin	NNMi Level 2 Operators
Mary	NNMi Level 2 Operators

12 User Account Mappings

User Groups

Name	Display Name
ChicagoLevel1	Chicago Level 1
ChicagoLevel2	Chicago Level 2
DetroitLevel1	Detroit Level 1
DetroitLevel2	Detroit Level 2
admin	NNMi Administrators
guest	NNMi Guest Users
level1	NNMi Level 1 Operators
level2	NNMi Level 2 Operators
client	NNMi Web Service Clients

7 User Groups

Create Security Groups

1. From the **Security Wizard** main page, select the **Map User Groups and Security Groups** option

2. Navigate to the **Security Groups** table.
3. Click ✳ **New**.
4. In the **Create Security Group** dialog box, enter the following:
    a. **Name**: Enter **Chicago Nodes**. The name must be a maximum of 255 characters. Alpha-numeric, spaces, and special characters (~ ! @ # $ % ^ &amp; * ( ) _+ -) are permitted.
    b. **Description**: Type a maximum of 2048 characters. Alpha-numeric, spaces, and special characters (~ ! @ # $ % ^ &amp; * ( ) _+ -) are permitted.
5. Click **Add**.
6. Repeat Step 4 and 5 to add the **Detroit Nodes**.
7. When you finish creating Security Groups, in the **Create Security Group** dialog box, click **Close** Close.
8. Continue or, click the **Save and Close** button to save your security configuration:

    Save & Close

    > **Note:** NNMi does not save any configuration changes until you click **Save and Close** to save your security configuration.

Map Security Groups to User Groups

1. From the **Security Wizard** main page, select the **Map User Groups and Security Groups** option.
2. Select **Chicago Nodes** in the **Security Groups** table.
3. In the **Security Group Mappings** drop-down selection box, select **Object Operator Level 2**.
4. In the **User Groups** table, click the ⇨ right arrow in the **ChicagoLevel2** row.

    The Security Group and User Group names appear in the **Security Group Mapping** table.
5. Repeat steps 2 through 4 to map the following User Groups and Security Groups:

    > **Tip:** Be sure to select the appropriate Object Access Privilege in the drop-down selection box under **Security Group Mappings**.

    ChicagoLevel1 User Group to the **Chicago Nodes**

    DetroitLevel1 User Group to the **Detroit Nodes**

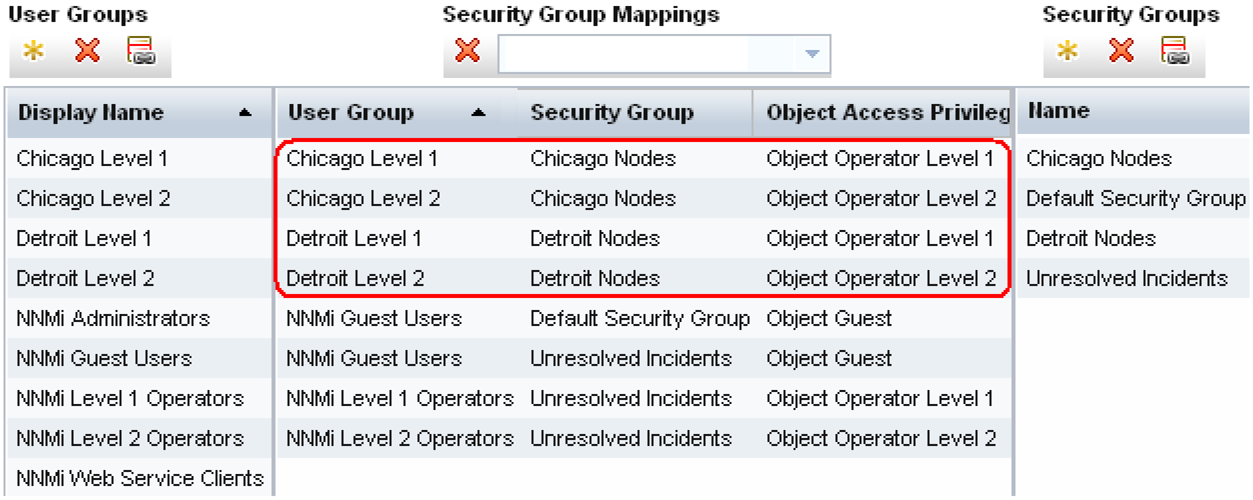
    DetroitLevel2 User Group to the **Detroit Nodes**
6. Continue or, click the **Save and Close** button to save your security configuration:

    Save & Close

    > **Note:** NNMi does not save any configuration changes until you click **Save and Close** to

save your security configuration.

Your Security Group to User Group mappings should look similar to the following:



User Groups

Display Name ▲
Chicago Level 1
Chicago Level 2
Detroit Level 1
Detroit Level 2
NNMi Administrators
NNMi Guest Users
NNMi Level 1 Operators
NNMi Level 2 Operators
NNMi Web Service Clients

Security Group Mappings

User Group ▲	Security Group	Object Access Privileg
Chicago Level 1	Chicago Nodes	Object Operator Level 1
Chicago Level 2	Chicago Nodes	Object Operator Level 2
Detroit Level 1	Detroit Nodes	Object Operator Level 1
Detroit Level 2	Detroit Nodes	Object Operator Level 2
NNMi Guest Users	Default Security Group	Object Guest
NNMi Guest Users	Unresolved Incidents	Object Guest
NNMi Level 1 Operators	Unresolved Incidents	Object Operator Level 1
NNMi Level 2 Operators	Unresolved Incidents	Object Operator Level 2

Security Groups

Name
Chicago Nodes
Default Security Group
Detroit Nodes
Unresolved Incidents

Assign the Nodes to the Appropriate Security Group

1. From the **Security Wizard** main page, select the **Assign Nodes to Security Groups** option.
2. Select a row in the **Security Groups** table.
3. In the **Available Nodes** table, do one of the following:
   a. Select a Node Group in the Node Group filter drop-down list to specify the nodes to be assigned to the Security Group.
   b. User Ctrl-Click to select each node you want to assign to the selected Security Group.
4. Click to specify that you want to assign the selected nodes to the Security Group.

   The **Nodes to be Assigned to Selected Group** table displays the list of nodes to be assigned to the selected Security Group.

   > **Note:** Out-of-box, NNMi assigns all Nodes the Default Security Group. See "Methods for Assigning Nodes to Security Groups" on page 563.

5. Repeat steps 2 through 4 to assign nodes to a selected Security Group.
6. Continue or, click the **Save and Close** button to save your security configuration:

   Save & Close

   > **Note:** NNMi does not save any configuration changes until you click Save and Close to save your security configuration.

View the Summary of Configuration Changes

From the **Security Wizard** main page, select the **View Summary of Changes** option.

NNMi displays a summary of the configuration changes made since you last saved your changes.

Save Your Configuration Changes

When you finish, click the **Save and Close** button to save your security configuration:

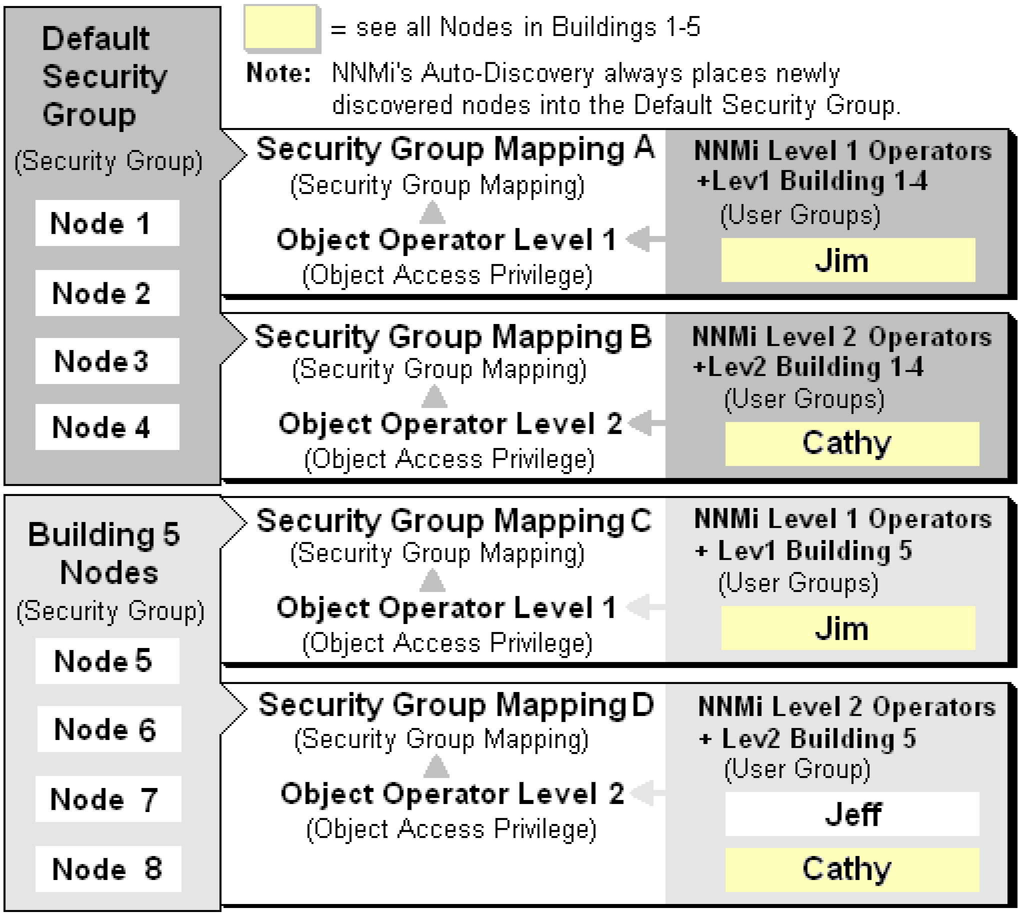
Note: NNMi does not save any configuration changes until you click **Save and Close** to save your security configuration.

Configure Security Example (Allow a Subset of Users to Access a Subset of Nodes)

This example uses NNMi's security configuration to allow a subset of users to access only those nodes in Building 5. The remaining users can access all nodes discovered by NNMi.

This location includes an NNMi Level 1 Operator (with less access privileges than Level 2 Operators) and an NNMi Level 2 Operator. Jeff is an NNMi Level 2 Operator who can access only the nodes in Building 5.

Note: Be sure to create a User Account that is mapped to the NNMi Administrator User Group so that one person has access to the Configuration workspace and all the nodes in the network. See "Restore the Administrator NNMi Role" on page 587for more information.



The following table lists the NNMi console access requirements (**NNMi User Group**[1]) and node access requirements (User Group, Object Access Privilege and Security Group) for each User Account.

> **Note:** You can place all operators into the NNMi Level 2 Operators if you want all operators to see all menu options, but only have the ability to run them based on their Object Access Privilege.

Example Security Configuration

User Accounts	NNMi User Groups	User Groups	Object Access Privileges	Security Groups
Jim	NNMi Level 1 Operators	Lev1Buildings1-4 Lev1Building5	Object Operator Level 1	Default Security Group

[1] NNMi User Groups are those User Groups provided by NNMi. Users cannot access the NNMi console until their User Account is mapped to at least one of the following NNMi User Groups: NNMi Administrators, NNMi Level 2 Operators, NNMi Level 1 Operators (with less access privileges than Level 2 Operators), and NNMi Guest Users

Example Security Configuration, continued

User Accounts	NNMi User Groups	User Groups	Object Access Privileges	Security Groups
Cathy	NNMi Level 2 Operators	Lev2Buildings1-4 Lev2Building5	Object Operator Level 2	Default Security Group
Jeff	NNMi Level 2 Operators	Lev2Building5	Object Operator Level 2	Building 5 Nodes

To set up security for this location follow these procedures:

- Remove the Default Security Group Mapping to NNMi User Groups: NNMi Level 1 Operators, NNMi Level 2 Operators, and NNMi Guest

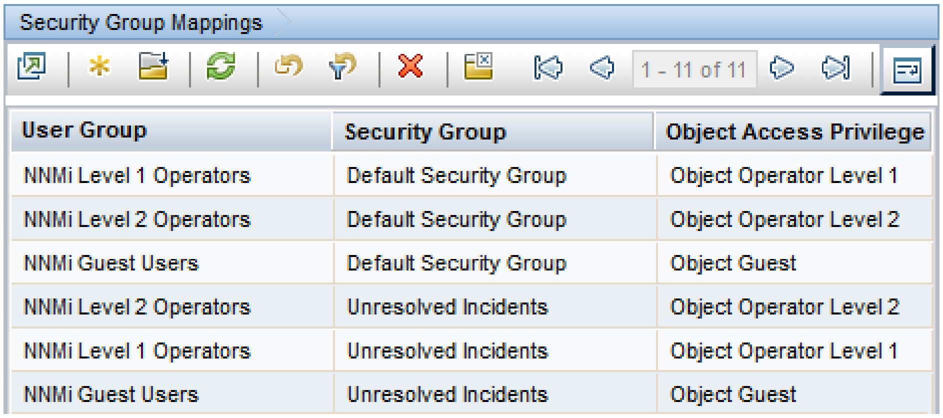
  Note: The **NNMi User Group**[1]s are provided for those NNMi administrators who are not concerned with Security configuration. After you remove these Security Group Mappings, the NNMi User Groups provide access to the NNMi console only rather than to the NNMi console and to all nodes.

- Create the User Accounts. (See the Example Security Configuration table.)
- Create Additional User Groups. (See the Example Security Configuration table.)
- Map User Accounts to NNMi User Groups. (See the Example Security Configuration table.)
- Create the Building 5 Security Group.
- Map each Security Group to the new User Groups. (See the Example Security Configuration table.)
- Assign the nodes to the appropriate Security Group.
- View a summary of your configuration changes

Remove the Default Security Group Mapping to NNMi User Groups

1. From the **Security Wizard** main page, select the **Map User Groups and Security Groups** option.
2. Navigate to the **Security Group Mappings** table.

[1] NNMi User Groups are those User Groups provided by NNMi. Users cannot access the NNMi console until their User Account is mapped to at least one of the following NNMi User Groups: NNMi Administrators, NNMi Level 2 Operators, NNMi Level 1 Operators (with less access privileges than Level 2 Operators), and NNMi Guest Users



User Group	Security Group	Object Access Privilege
NNMi Level 1 Operators	Default Security Group	Object Operator Level 1
NNMi Level 2 Operators	Default Security Group	Object Operator Level 2
NNMi Guest Users	Default Security Group	Object Guest
NNMi Level 2 Operators	Unresolved Incidents	Object Operator Level 2
NNMi Level 1 Operators	Unresolved Incidents	Object Operator Level 1
NNMi Guest Users	Unresolved Incidents	Object Guest

3. Click the row representing the **NNMi Level 1 Operators** User Group.

4. Click the ✖ Delete icon to remove the Default Security Group to NNMi Level 1 Operators User Group mapping.

5. Repeat steps 3 and 4 to remove the Default Security Group to **NNMi Level 2 Operator** and the **NNMi Guest** User Group mappings.

6. Continue or, click the **Save and Close** button to save your security configuration:

   Save & Close

> **Note:** NNMi does not save any configuration changes until you click **Save and Close** to save your security configuration.

Create User Accounts

1. From the **Security Wizard** main page, select the **Map User Accounts and User Groups** option.

2. Navigate to the **User Accounts** table.

3. Click ✳ **New**.

4. In the **Create User Account** dialog box, enter the following:

   a. **Name**: Enter the user name **Jim**.

   b. **Directory Service**: Do not enable this option.

      □ = User name and password are stored in the NNMi database. See "NNMi Configuration Settings to Control NNMi Access" on page 505.

   c. **Password**: Enter the Password value **Jim**. The Password value can be up to 40 alpha-numeric characters, punctuation, spaces, and underline characters.

5. Click **Add**.

6. Repeat steps 4 and 5 to add each User Account. (See the Example Security Configuration table.)

7. When you finish creating User Accounts, in the **Create User Account** dialog box, click **Close** Close.

8. Continue or, click the **Save and Close** button to save your security configuration:

   > **Note:** NNMi does not save any configuration changes until you click **Save and Close** to save your security configuration.

Create Additional User Groups

1. From the **Security Wizard** main page, select the **Map User Accounts and User Groups** option
2. Navigate to the **User Groups** table.
3. Click * **New**.
4. In the **Create User Group** dialog box, enter the following:
   a. **Name**: Enter **Lev1Building1-4**. The name can be a maximum of 40 alpha-numeric characters. Spaces are not permitted.
   b. **Display Name**: Enter **Lev1 Building 1-4**. The Display Name is displayed in the NNMi console to identify this User Group. Enter a maximum of 50 characters. Alpha-numeric, spaces, and special characters (~ ! @ # $ % ^ &amp; * ( ) _+ -) are permitted.

      > **Tip:**

   c. **Directory Service Name**: When a directory service defines this User Group, enter the group’s Distinguished Name. This example does not use this option. NNMi communicates with the directory service using Lightweight Directory Access Protocol (LDAP).
      - "Lightweight Directory Access Protocol (LDAP) to Control NNMi Access" on page 506.
      - "X.509 Certificates to Control NNMi Access" on page 507
   d. **Description**: Type a maximum of 2048 characters to describe this User Group. Alpha-numeric, spaces, and special characters (~ ! @ # $ % ^ &amp; * ( ) _+ -) are permitted.
5. Click **Add**.
6. Repeat steps 4 and 5 to add the **Lev1Building5**, **Lev2Building1-4**, and **Lev2Building5** User Groups. (See the Example Security Configuration table.)
7. When you finish creating User Groups, in the **Create User Group** dialog box, click **Close**

   .
8. Continue, or click the **Save and Close** button to save your security configuration:

   > **Note:** NNMi does not save any configuration changes until you click **Save and Close** to save your security configuration

Map User Accounts to User Groups

> **Note:** A User Account cannot access the NNMi console until after it is mapped to one of the NNMi User Groups.

1. From the **Security Wizard** main page, select the **Map User Accounts and User Groups** option
2. Select **Jim** in the **User Accounts** table.
3. In the **User Groups** table, select the left arrow that precedes the **NNMi Level 1 Operators** User Group.

   The User Account and User Group names appear in the **User Account Mapping** table.
4. Repeat steps 2 and 3 to assign each User Account to the appropriate User Group. (See the Example Security Configuration table.):

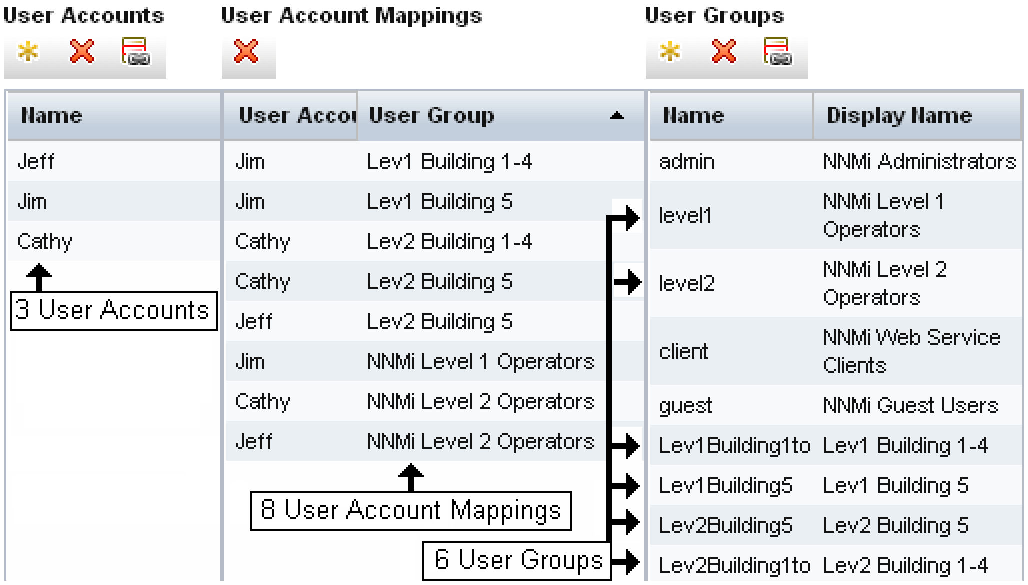
   Assign **Jim** to the **Lev1Building1-4** and **Lev1Building5** User Group

   Assign **Cathy** to the **NNMi Level 2 Operators**, **Lev2Building1-4**, and **Lev2Building5** User Groups

   Assign **Jeff** to the **NNMi Level 2 Operators** and **Lev2Building 5** User Groups.
5. Continue or, click the **Save and Close** button to save your security configuration:

   > **Note:** NNMi does not save any configuration changes until you click **Save and Close** to save your security configuration.

   Your User Account to User Group mappings should look similar to the following:



User Accounts

Name
Jeff
Jim
Cathy

3 User Accounts

User Account Mappings

User Accou	User Group
Jim	Lev1 Building 1-4
Jim	Lev1 Building 5
Cathy	Lev2 Building 1-4
Cathy	Lev2 Building 5
Jeff	Lev2 Building 5
Jim	NNMi Level 1 Operators
Cathy	NNMi Level 2 Operators
Jeff	NNMi Level 2 Operators

8 User Account Mappings

User Groups

Name	Display Name
admin	NNMi Administrators
level1	NNMi Level 1 Operators
level2	NNMi Level 2 Operators
client	NNMi Web Service Clients
guest	NNMi Guest Users
Lev1Building1to	Lev1 Building 1-4
Lev1Building5	Lev1 Building 5
Lev2Building5	Lev2 Building 5
Lev2Building1to	Lev2 Building 1-4

6 User Groups

Create the Building 5 Security Group

1. From the **Security Wizard** main page, select the **Map User Groups and Security Groups** option
2. Navigate to the **Security Groups** table.
3. Click ✳ **New**.
4. In the **Create Security Group** dialog box, enter the following:
   a. **Name**: Enter **Building 5 Nodes**. The name must be a maximum of 255 characters. Alpha-numeric, spaces, and special characters (~ ! @ # $ % ^ &amp; * ( ) _+ -) are permitted.
   b. **Description**: Type a maximum of 2048 characters. Alpha-numeric, spaces, and special characters (~ ! @ # $ % ^ &amp; * ( ) _+ -) are permitted.
5. Click **Add**.
6. When you finish creating Security Groups, in the **Create Security Group** dialog box, click **Close** Close.
7. Continue, or click the **Save and Close** button to save your security configuration:

   > **Note:** NNMi does not save any configuration changes until you click **Save and Close** to save your security configuration.

Map User Groups to Security Groups

1. From the **Security Wizard** main page, select the **Map User Groups and Security Groups** option.
2. Select **Default Security Group** in the **Security Groups** table.
3. In the **Security Group Mappings** drop-down selection box, select **Object Operator Level 1**.
4. In the **User Groups** table, click the ⇨ right arrow in the **Lev1Building1-4** row.

   The Security Group and User Group names appear in the **Security Group Mapping** table.
5. Repeat steps 2 through 4 to assign the following Security Group Mappings:

   > **Tip:** Be sure to select the appropriate Object Access Privilege in the drop-down selection box under **Security Group Mappings**.

   Lev1Building5 User Group to the **Building 5 Nodes**.

   Lev2Building1-4 User Group to the **Default Security Group**

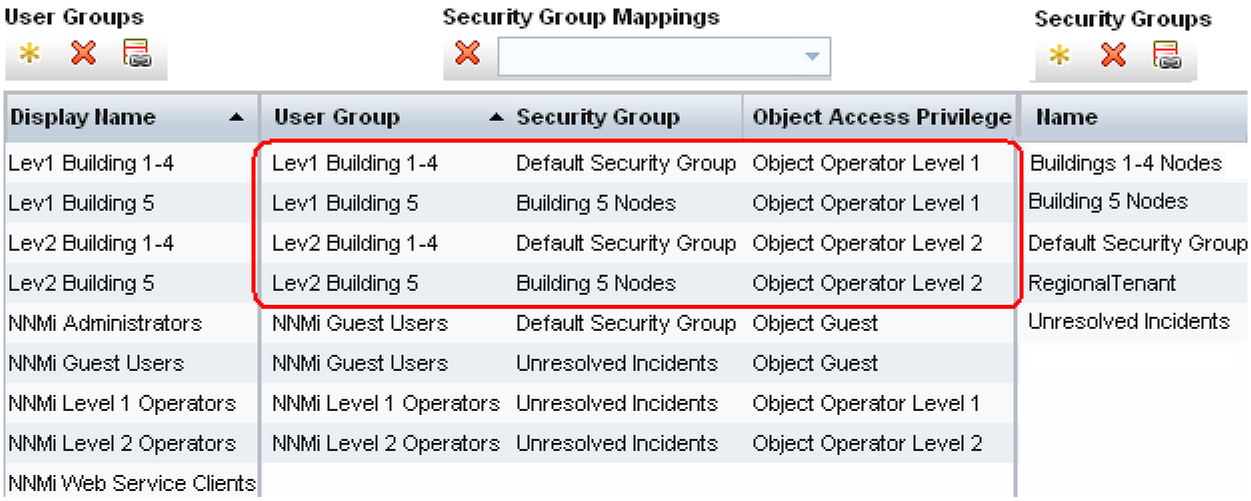
   Lev2Building5 User Group to the **Building 5 Nodes**.
6. Continue or, click the **Save and Close** button to save your security configuration:

   Save & Close

   > **Note:** NNMi does not save any configuration changes until you click **Save and Close** to

> save your security configuration.

Your User Group to Security Group mappings should look similar to the following:



Display Name
Lev1 Building 1-4
Lev1 Building 5
Lev2 Building 1-4
Lev2 Building 5
NNMi Administrators
NNMi Guest Users
NNMi Level 1 Operators
NNMi Level 2 Operators
NNMi Web Service Clients

User Group	Security Group	Object Access Privilege
Lev1 Building 1-4	Default Security Group	Object Operator Level 1
Lev1 Building 5	Building 5 Nodes	Object Operator Level 1
Lev2 Building 1-4	Default Security Group	Object Operator Level 2
Lev2 Building 5	Building 5 Nodes	Object Operator Level 2
NNMi Guest Users	Default Security Group	Object Guest
NNMi Guest Users	Unresolved Incidents	Object Guest
NNMi Level 1 Operators	Unresolved Incidents	Object Operator Level 1
NNMi Level 2 Operators	Unresolved Incidents	Object Operator Level 2

Name
Buildings 1-4 Nodes
Building 5 Nodes
Default Security Group
RegionalTenant
Unresolved Incidents

Assign the Nodes to the Appropriate Security Group

1. From the **Security Wizard** main page, select the **Assign Nodes to Security Groups** option.
2. Select the **Building 5 Nodes** row in the **Security Groups** table.
3. In the **Available Nodes** table, do one of the following:
   a. Select a Node Group in the Node Group filter drop-down list to specify the nodes to be assigned to the Security Group.
   b. User Ctrl-Click to select each node you want to assign to the **Building 5 Nodes**.
4. Click to specify that you want to assign the selected nodes to the Security Group.

   The **Nodes to be Assigned to Selected Group** table displays the list of nodes to be assigned to the selected Security Group.
5. Repeat steps 2 through 4 to assign nodes to a selected Security Group.
6. Continue or, click **Save and Close** to save your security configuration:

> **Note:** NNMi does not save any configuration changes until you click **Save and Close** to save your security configuration.

View the Summary of Configuration Changes

From the **Security Wizard** main page, select the **View Summary of Changes** option.

NNMi displays a summary of the configuration changes made since you last saved your changes.

Save Your Configuration Changes

When you finish, click the **Save and Close** button to save your security configuration:

> **Note:** NNMi does not save any configuration changes until you click **Save and Close** to save your security configuration.

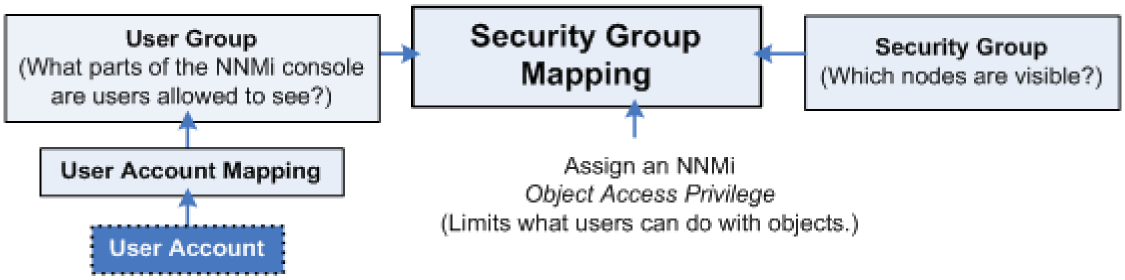
User Account Tasks

NNMi administrators can configure User Accounts using the following methods:

- The Configuration Wizard ("Create and Delete User Accounts Using the Security Wizard" on page 545)
- The User Accounts view ("Configure User Accounts (User Account Form)" below)
- The nnmsecurity.ovpl command line tool

Configure User Accounts (User Account Form)

NNMi User Account configurations provide NNMi user name and password settings, as well as indicate whether NNMi should use an external resource for password information. See "About User Accounts" on page 513.


User Group
(What parts of the NNMi console are users allowed to see?)
Security Group Mapping
Security Group
(Which nodes are visible?)
User Account Mapping
User Account
Assign an NNMi
Object Access Privilege
(Limits what users can do with objects.)

> **Tip:** NNMi administrators can also use the Security Wizard or command line to complete this task. See "Create and Delete User Accounts Using the Security Wizard" on page 545 or nnmsecurity.ovpl.

To configure NNMi user names and passwords use the following instructions:

1. Navigate to the **User Accounts** view.
   a. From the workspaces navigation panel, select the **Configuration** workspace.
   b. Expand **Security**.
   c. Select **User Accounts**.

   > **Tip:** You can filter the User Accounts table view by User Group or Security Group.

2. Do one of the following:

- To create a new configuration, click the ✳ **New** icon.
- To edit an existing configuration, double-click the User Account definition you want to edit.
- To delete a User Account, see "Delete a User Account" on the next page.

3. Make your configuration choices. See the User Account Attributes table.

4. Click **Save and Close** to save your changes and return to the **User Accounts** view.

   > **Note:** You must click **Save and Close** to save your changes each time you create a User Account.

5. NNMi users can belong to more than one User Group. You must assign each User Account to a preconfigured User Group provided by NNMi before that user can access NNMi. See "User Groups Provided in NNMi" on page 547 and for more information.

User Account Attributes

Attribute	Description
Name	Enter the user name to be assigned to this user.
Directory Service Account	□ = User name and password are stored in the NNMi database. See "NNMi Configuration Settings to Control NNMi Access" on page 505. ☑ = NNMi uses Lightweight Directory Access Protocol (LDAP) or X509 Certificates such as Public Key Infrastructure (PKI) user authentication. Additional steps are required. See the following topics: • "Lightweight Directory Access Protocol (LDAP) to Control NNMi Access" on page 506. • "X.509 Certificates to Control NNMi Access" on page 507
Password	Enter the **Password** value. Type up to 40 alpha-numeric characters, punctuation, spaces, and underline characters. **Note:** If you enabled **Directory Service Account** ☑, do not provide a Password. **Tip:** When NNMi is configured with **Directory Service Account** □, NNMi users who are assigned to the following Security Group Mapping can change their NNMi password at any time using **File → Change Password.** *Object Access Privilege* = one of the following: • Object Administrator • Object Operator Level 2 • Object Operator Level 1 (with less access privileges than Level 2)
	Re-type the **Password** value.

Related Topics:

"Delete a User Account" below

"Change Password, Name" on the next page

"Restore the Administrator NNMi Role" on page 587

Delete a User Account

To deny a user's access to the NNMi console, delete their user configuration settings from the NNMi database.

> **Note:** Ignore this topic if NNMi is configured to access LDAP information for user group assignments. When NNMi is configured in that way, to disable a user's access to NNMi, you must use the appropriate process required by your environment's directory service software (see "Lightweight Directory Access Protocol (LDAP) to Control NNMi Access" on page 506 and "X.509 Certificates to Control NNMi Access" on page 507).

> **Caution:** If you delete the last NNMi user assigned to the NNMi Administrators User Group, no one can access the Configuration workspace. See "Restore the Administrator NNMi Role" on page 587 for more information about how to recover from this mistake.

> **Tip:** NNMi administrators can also use the Security Wizard or command line to complete this task. See "Create and Delete User Accounts Using the Security Wizard" on page 545 or nnmsecurity.ovpl.

To deny a user's access to NNMi:

1. Navigate to the **User Accounts** view.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Expand **Security**.
   c. Select **User Accounts**.
2. Select the row containing the User Account you want to delete.
3. Click the ✖ Delete icon.

   The user's configuration is automatically removed from the User Accounts view.

> **Note:** If you remove the User Account for a user who is currently signed into the NNMi console, the change does not take effect until the next time the user signs in. By default, the NNMi timeout limit is 18 hours. If a user has not signed out within 18 hours, NNMi forces the user to sign out. To change the Console Timeout value, see "Configuring the NNMi User Interface" on page 467

> **Tip:** Access the Incident Browsing workspace. Open the All Incidents view. Sort this view

using the Assigned To (AT) column. Reassign all Incidents associated with any user you deleted (see Assign an Incident).

Change Password, Name

Only NNMi administrators can add and delete accounts and change NNMi User Accounts and User Groups.

- If configuring NNMi to store user names and passwords in the NNMi database, use the following instructions.
- If configuring NNMi to use an external User Authentication Method (passwords stored outside of the NNMi database), see "Lightweight Directory Access Protocol (LDAP) to Control NNMi Access" on page 506 or "X.509 Certificates to Control NNMi Access" on page 507.

Tip: NNMi administrators can also use the Security Wizard or command line to complete this task. See "Create and Delete User Accounts Using the Security Wizard" on the next page or nnmsecurity.ovpl.

To change an NNMi user name:

You must "Delete a User Account" on the previous page, and then recreate the account mapping (see "NNMi Configuration Settings to Control NNMi Access" on page 505).

To change an NNMi password:

Note: If you are not using Lightweight Directory Access Protocol (LDAP) or X.509 Certificates to manage NNMi users, User Accounts assigned to the following User Groups can change their password using **File** → **Change Password**: NNMi Administrators, NNMi Level 2 Operators, and NNMi Level 1 Operators (with less access privileges than Level 2 Operators). See Change Your Password for more information.

1. Navigate to the **User Accounts** view.
   a. From the Workspaces navigation panel, select the **Configuration** workspace.
   b. Expand **Security**.
   c. Select **User Accounts**.
2. Double-click the row representing the account you want to edit.
3. Locate the **Password** attribute and edit the **Password** value. Type up to 40 alpha-numeric characters, punctuation, spaces, and underline characters.
4. Retype the new password.
5. Click **Save and Close**. NNMi immediately implements your changes.

To change an NNMi User Group to User Account assignment:

Note: To change a User Group to User Account assignment, you first delete the User Account

> mapping. If you change the User Account or User Group configuration for a user who is currently signed into the NNMi console, the change does not take effect until the next time the user signs in. By default, the NNMi timeout limit is 18 hours. If a user has not signed out within 18 hours, NNMi forces the user to sign out. To change the Console Timeout value, see "Configuring the NNMi User Interface" on page 467

1. Navigate to the **User Account Mappings** view.
   a. From the Workspaces navigation panel, select the **Configuration** workspace.
   b. Expand **Security**.
   c. Select **User Account Mappings**.
2. Select the row representing the User Account mapping you want to change.
3. Delete the User Account mapping by clicking the ✖ Delete icon.
4. Select the ✳ New icon to configure the new User Account mapping.
5. Make your configuration choices. (See the User Account Mapping Attributes table.)
6. Click **Save and Close**.

User Account Mapping Attributes

Attribute	Description
User Group	In the **User Group** attribute, click the **Lookup** icon. ▪ To create new User Group, click the ✳ **New** icon and provide the required information. (See "Configure User Groups (User Group Form)" on page 550 for more information.) ▪ To select an NNMi User Group configuration, click the **Quick Find** icon and make a selection.
User Account	In the **User Account** attribute, click the **Lookup** icon. ▪ To create new User Account, click the ✳ **New** icon and provide the required information. See "Configure User Accounts (User Account Form)" on page 541 for more information.) ▪ To select an NNMi User Group configuration, click the **Quick Find** icon and make a selection. **Note:** If you map a User Account to two or more NNMi User Groups, NNMi gives the User Account the privileges associated with each mapped NNMi User Group.

Create and Delete User Accounts Using the Security Wizard

For more information about User Accounts, see "About User Accounts" on page 513.

> **Tip:** NNMi administrators can also use the User Accounts view or command line to complete this task. See "Configure User Accounts (User Account Form)" on page 541 or nnmsecurity.ovpl.

To create a User Account:

1. From the **Security Wizard** main page, select the **Map User Accounts and User Groups** option.
2. Navigate to the **User Accounts** table.
3. Click ✱ **New**.
4. In the **Create User Account** dialog box, enter the following:
   a. **Username**: Enter the user name to be assigned to this user.
   b. **Password**: Enter the Password value. Type up to 40 alpha-numeric characters, punctuation, spaces, and underline characters.

   > **Note:** The Security Wizard is unable to create accounts for use with LDAP or PKI user authentication. These accounts may be created using the User Accounts Form or the `nnmsecurity.ovpl` command. See "Configure User Accounts (User Account Form)" on page 541.

5. Click **Add**.
6. Repeat Step 4 and 5 to add each User Account.
7. When you finish adding User Accounts in the **Create User Account** dialog box, click **Close** [Close].
8. When you finish your security configuration, click **Save and Close** to save your security configuration.

To modify a User Account: see "Change Password, Name" on page 544.

To delete a User Account:

1. Select a row in the **User Accounts** table.
2. Click ✖ **Delete**.
3. When you finish, click the **Save and Close** button to save your security configuration:

   [Save & Close]

   > **Caution:** If you remove the User Account for a user who is currently signed into the NNMi console, the change does not take effect until the next time the user signs in. By default, the NNMi timeout limit is 18 hours. If a user has not signed out within 18 hours, NNMi forces the user to sign out. To change the Console Timeout value, see "Configuring the NNMi User Interface" on page 467

> Access the Incident Browsing workspace. Open the All Incidents view. Sort this view using the Assigned To (AT) column. Reassign all Incidents associated with any user you deleted (see Assign an Incident).

NNMi User Accounts can be assigned to one or more User Groups. You must assign each User Account to one of the following NNMi User Groups so users can access the NNMi console:

- NNMi Administrators
- NNMi Level 2 Operators
- NNMi Level 1 Operators (with less access privileges than Level 2 Operators)
- NNMi Guest Users

See "Assign User Accounts to User Groups Using the Security Wizard Page" on page 558 for more information.

User Group Tasks

NNMi administrators can configure User Groups using the following methods:

- The Configuration Wizard ("Create and Delete User Groups Using the Security Wizard" on page 552)
- The User Accounts view ("Configure User Groups (User Group Form)" on page 550)
- The nnmsecurity.ovpl command line tool

User Groups Provided in NNMi

When the NNMi administrator configures NNMi Security, each User Account must be mapped to one or more User Group.

The following predefined **NNMi User Group**[1]s determine the NNMi user's access to the NNMi console workspaces, forms, and actions. Each User Account must be mapped to one of these predefined NNMi User Groups before users can access the NNMi console:

- NNMi Administrators
- NNMi Level 2 Operators
- NNMi Level 1 Operators (with less access privileges than Level 2 Operators)
- NNMi Guest Users

[1] NNMi User Groups are those User Groups provided by NNMi. Users cannot access the NNMi console until their User Account is mapped to at least one of the following NNMi User Groups: NNMi Administrators, NNMi Level 2 Operators, NNMi Level 1 Operators (with less access privileges than Level 2 Operators), and NNMi Guest Users

Note: NNMi provides two additional User Groups:

- NNMi Global Operators (*secondary*)

  Assigning users to this *secondary* group, in addition to the user's currently assigned NNMi Guest User, NNMi Level 1 Operator, or NNMi Level 2 Operator assignment, provides access to all topology objects, but does not change any other aspect of their currently assigned NNMi Guest User, NNMi Level 1 Operator, or NNMi Level 2 Operator assignment.

  Users assigned to the NNMi Administrators User Group do not need any *secondary* group assignment. These users already can access all topology objects.

- NNMi Web Services Client

  Used *only to provide access for software* that is integrated with NNMi. See "Integrations with HP and Third-Party Products" on page 1487 - for example, "HP RAMS MPLS WAN Configuration (NNMi Advanced)" on page 1410). Do not use any other User Group for software integrations.

You cannot delete these predefined NNMi User Groups.

If you map a User Account to two or more NNMi User Groups, NNMi gives the User Account the privileges associated with each User Group to which the User Account is assigned.

Note: NNMi administrators can also create User Groups. Creating User Groups enables you to fine tune User Group access when using Security Groups. For example, you might want one User Group to have Level 2 Operator access to the nodes in one Security Group and Level 1 Operator access to nodes in another Security Group. See "Configure User Groups (User Group Form)" on page 550 and "Configure Security Groups (Security Group Form)" on page 560 for more information.

For details about User Groups, see the following topics:

- "Determine which NNMi User Group to Assign" on the next page (controls access to views and forms)
- "Control Menu Access" on page 574 (NNMi administrators control which User Groups can access a subset of Action menu items)
- "Configure Basic Settings for a Node Group Map" on page 489 (For each Node Group Map, the **Minimum NNMi Role for Saving Map Layout** attribute setting controls the minimum **NNMi Role**[1] required for saving the layout after the user repositions nodes on the map. The NNMi Role is assigned to a User Account through the NNMi User Group.)

[1] Determined by your membership in one of four special NNMi User Groups (NNMi Administrators, NNMi Level 2 Operators, NNMi Level 1 Operators, or NNMi Guest Users). This membership determines what you can see and do within the NNMi console.

Determine which NNMi User Group to Assign

Before configuring NNMi sign-in access for your team, determine which predefined **NNMi User Group**[1] is appropriate for each team member. The User Groups are hierarchical, meaning the higher level User Groups include all privileges of the lower level User Groups in the hierarchy (Administrator is highest, Guest is lowest).

Note: NNMi provides a special `Web Services Client` User Group used *only to provide access for software* that is integrated with NNMi (for example, see "HP RAMS MPLS WAN Configuration (NNMi Advanced)" on page 1410). Do not use any other User Group for software integrations.

As NNMi administrator, you can change the following aspects of User Group definitions:

- "Control Menu Access" on page 574 (restrict access to certain NNMi Actions menu items and Tools menu items - provide tighter security than those enforced by the default settings.) See also "Configure Launch Actions" on page 1422 for more information about adding options to the NNMi Actions menu.
- "Configure Basic Settings for a Node Group Map" on page 489. (For each Node Group Map, the **Minimum NNMi Role for Saving Map Layout** attribute setting controls the minimum User Group required for saving the layout after the user repositions nodes on the map. The **NNMi Role**[2] is assigned to a User Account through the NNMi User Group.
- "Set Up Command Line Access to NNMi" on page 577 (Use to control access to NNMi command line commands.)

The following table lists the User Group required to access NNMi workspaces. You cannot modify User Group settings for workspaces. See About Workspaces for more information about workspaces. See Views Provided by NNMi for more information about the views provided in each workspace.

Access to Workspaces

Workspaces	NNMi Guest Users	NNMi Level 1 Operators	NNMi Level 2 Operators	NNMi Administrators
All views in the Incident workspaces	Yes	Yes	Yes	Yes
All views in the Topology workspace	Yes	Yes	Yes	Yes
All views in the Monitoring	Yes	Yes	Yes	Yes

[1] NNMi User Groups are those User Groups provided by NNMi. Users cannot access the NNMi console until their User Account is mapped to at least one of the following NNMi User Groups: NNMi Administrators, NNMi Level 2 Operators, NNMi Level 1 Operators (with less access privileges than Level 2 Operators), and NNMi Guest Users

[2] Determined by your membership in one of four special NNMi User Groups (NNMi Administrators, NNMi Level 2 Operators, NNMi Level 1 Operators, or NNMi Guest Users). This membership determines what you can see and do within the NNMi console.

Access to Workspaces , continued

Workspaces	NNMi Guest Users	NNMi Level 1 Operators	NNMi Level 2 Operators	NNMi Administrators
workspace				
All views in the Troubleshooting workspace	Yes	Yes	Yes	Yes
All views in the Inventory workspace	Yes	Yes	Yes	Yes
All views in the Management Mode workspace			Yes	Yes
All views in the Configuration workspace				Yes

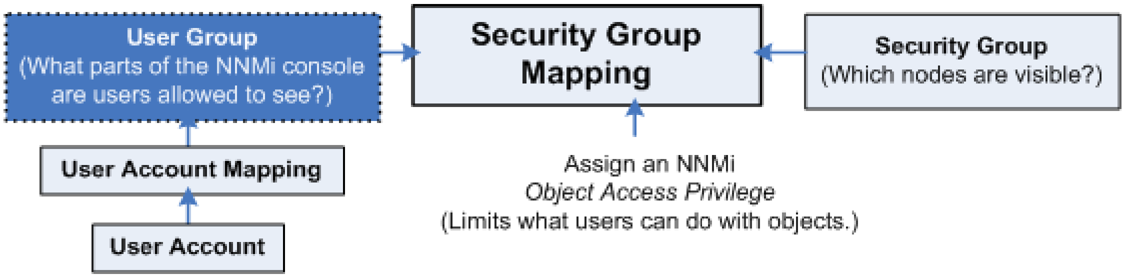
The following table provides some examples of how NNMi User Groups control permission for modifications to certain forms. You cannot modify User Group settings for forms.

Access to Forms (some examples)

Forms	NNMi Guest Users	NNMi Level 1 Operators	NNMi Level 2 Operators	NNMi Administrators
Node forms	Read-Only	Read-Write except Management Mode field which is Read-Only	Read-Write	Read-Write
Interface forms	Read-Only	Read-Write except Management Mode field which is Read-Only	Read-Write	Read-Write
IP Address forms	Read-Only	Read-Write except Management Mode field which is Read-Only	Read-Write	Read-Write
IP Subnet forms	Read-Only	Read-Write except Management Mode field which is Read-Only	Read-Write	Read-Write
Incident forms	Read-Only	Read-Write	Read-Write	Read-Write
Node Group forms	Read-Only	Read-Only	Read-Only	Read-Write
Configuration Forms				Read-Write

Configure User Groups (User Group Form)

Use this User Group form to establish any User Groups required for your NNMi Security strategy. See "Determine Your Security Strategy" on page 507.


User Group
(What parts of the NNMi console are users allowed to see?)
Security Group Mapping
Security Group
(Which nodes are visible?)
User Account Mapping
User Account
Assign an NNMi
Object Access Privilege
(Limits what users can do with objects.)

Each NNMi user must belong to at least one predefined **NNMi User Group**[1]. See "User Groups Provided in NNMi" on page 547 and "Determine which NNMi User Group to Assign" on page 549. These predefined NNMi User Groups cannot be deleted.

Each NNMi user can belong to one or more User Groups that the NNMi administrators create. See "About User Groups" on page 513.

> **Tip:** NNMi administrators can also use the Security Wizard or command line to complete this task. See "Create and Delete User Groups Using the Security Wizard" on the next page or nnmsecurity.ovpl.

To configure a User Group, do the following:

1. Navigate to the **User Groups** view.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Expand **Security**.
   c. Select **User Groups**.

   > **Tip:** You can filter the User Groups table view by Security Group.

2. Do one of the following:
   - To create a new configuration, click the ✱ **New** icon.
   - To edit an existing configuration, double-click the User Groups definition you want to edit.
   - To delete an existing configuration, click the ✖ **Delete** icon.
3. Make your configuration choices. (See the User Group Attributes table.)
4. Make your additional configuration choices. Click here for a list of choices .
5. Click **Save and Close** to apply your changes.

[1] NNMi User Groups are those User Groups provided by NNMi. Users cannot access the NNMi console until their User Account is mapped to at least one of the following NNMi User Groups: NNMi Administrators, NNMi Level 2 Operators, NNMi Level 1 Operators (with less access privileges than Level 2 Operators), and NNMi Guest Users

User Group Attributes

Attribute	Description
Name	Enter the name that uniquely identifies the User Group. Enter a maximum of 40 alpha-numeric characters. Spaces are not permitted.
Display Name	Enter the name that should be displayed in the NNMi console to identify this User Group. Enter a maximum of 50 characters. Alpha-numeric, spaces, and special characters (~ ! @ # $ % ^ & * ( ) _+ -) are permitted.
Directory Service Name	*Optional*. When Lightweight Directory Access Protocol (LDAP) defines this User Group, enter the group's Distinguished Name. See the following topics: • "Lightweight Directory Access Protocol (LDAP) to Control NNMi Access" on page 506. • "X.509 Certificates to Control NNMi Access" on page 507
Description	Type a maximum of 2048 characters to describe this User Group. Alpha-numeric, spaces, and special characters (~ ! @ # $ % ^ & * ( ) _+ -) are permitted.

Create and Delete User Groups Using the Security Wizard

For more information about User Accounts, see "About User Groups" on page 513.

> **Tip:** NNMi administrators can also use the User Groups view or command line to complete this task. See "Configure User Groups (User Group Form)" on page 550 or nnmsecurity.ovpl.

To create User Groups:

1. From the **Security Wizard** page, do one of the following:
   a. Select the **Map User Accounts and Security Groups** option.
   b. Select the **Map User Groups and Security Groups** option.
2. Navigate to the **User Groups** table.
3. Click ✱ **New**.
4. In the **Create User Group** dialog box, enter the following:
   a. **Name**: Enter the name that uniquely identifies the User Group. Enter a maximum of 40 alpha-numeric characters. Spaces are not allowed.
   b. **Display Name**: Enter the name that should be displayed in the NNMi console to identify this User Group. Enter a maximum of 50 characters. Alpha-numeric, spaces, and special characters (~ ! @ # $ % ^ &amp; * ( ) _+ -) are permitted.
   c. **Directory Service Name**: Optional. When a directory service defines this User Group, enter the group's Distinguished Name. NNMi communicates with the directory service using Lightweight Directory Access Protocol (LDAP). See one of the following topics:

- "Lightweight Directory Access Protocol (LDAP) to Control NNMi Access" on page 506.
- "X.509 Certificates to Control NNMi Access" on page 507

d. **Description**: Type a maximum of 2048 characters to describe this User Group. Alpha-numeric, spaces, and special characters (~ ! @ # $ % ^ &amp; * ( ) _+ -) are permitted.

5. Click **Add**.
6. Repeat Step 4 and 5 to add each User Group.
7. When you finish adding User Groups,in the **Create User Group** dialog box, click **Close**

.
8. When you finish, click the **Save and Close** button to save your security configuration:

> **Note:** NNMi does not save any configuration changes until you click **Save and Close** to save your security configuration.

To delete User Groups:

1. Select a row in the **User Groups** table.
2. Click ✖**Delete**.
3. When you finish, click the **Save and Close** button to save your security configuration:

> **Note:** NNMi does not save any configuration changes until you click **Save and Close** to save your security configuration.

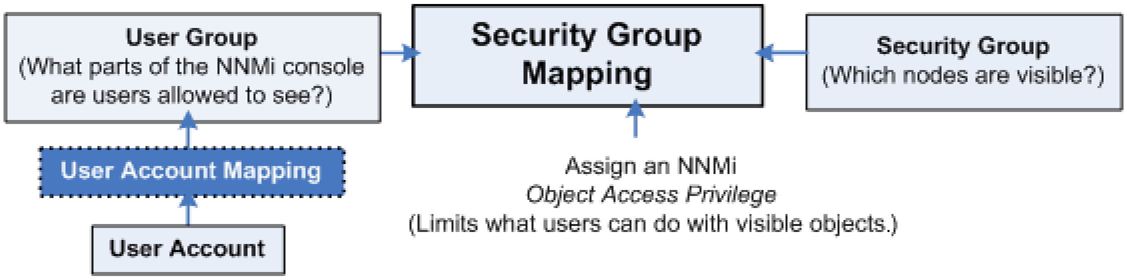
User Account Mapping Tasks

NNMi administrators can map User Accounts to User Groups using the following methods:

- The Configuration Wizard ("Map User Accounts and User Groups " on page 556)
- The User Account Mappings view ("Map User Accounts to User Groups (User Account Mapping Form)" below)
- The nnmsecurity.ovpl command line tool

Map User Accounts to User Groups (User Account Mapping Form)

To assign User Accounts to User Groups use the following instructions. See "About User Account Mappings" on page 514.


User Group
(What parts of the NNMi console are users allowed to see?)
Security Group Mapping
Security Group
(Which nodes are visible?)
User Account Mapping
Assign an NNMi
Object Access Privilege
(Limits what users can do with visible objects.)
User Account

The NNMi administrator must assign each User Account to a predefined NNMi User Group before that user can access NNMi. See "User Groups Provided in NNMi" on page 547 for more information.

> **Tip:** NNMi administrators can also use the Security Wizard and to complete this task. See "Map User Accounts and User Groups " on page 556.

To assign a User Account to a User Group:

1. Navigate to the **User Accounts Mappings** view:
   a. From the workspaces navigation panel, select the **Configuration** workspace.
   b. Expand **Security**.
   c. Select **User Account Mappings**.
2. Do one of the following:
   - To create a new configuration, click the ✱ **New** icon, and continue.
   - To edit an existing configuration, double-click the Mappings definition you want to edit, and continue.
   - To delete a Mapping, see "Delete a User Account" on page 543.
3. Make your configuration choices. See the User Account Mapping Attributes table.

   > **Tip:** NNMi can be configured to use the Lightweight Directory Access Protocol (LDAP) and X.509 Certificates such as Public Key Infrastructure (PKI) user authentication for NNMi user names, passwords, and User Group Membership assignments. Additional steps are required. See "Choose a Mode for NNMi Access" on page 504.

4. Click the **Save and Close** icon to save your changes and return to the **User Accounts Mappings** view.

   > **Note:** If you create a User Account to User Group mapping for an NNMi user who is currently signed into the NNMi console, the change does not take effect until the next time the user signs in. By default, the NNMi timeout limit is 18 hours. If a user has not signed out within 18 hours, NNMi forces the user to sign out. To change the Console Timeout value, see "Configuring the NNMi User Interface" on page 467

User Account Mapping Attributes

Attribute	Description
User Group	In the **User Group** attribute, click the **Lookup** icon. • To create new User Group, click the **New** icon and provide the required information. (See "Configure User Groups (User Group Form)" on page 550 for more information.) • To select an NNMi User Group configuration, click the **Quick Find** icon and make a selection.
User Account	In the **User Account** attribute, click the **Lookup** icon. • To create new User Account, click the **New** icon and provide the required information. See "Configure User Accounts (User Account Form)" on page 541 for more information.) • To select an NNMi User Group configuration, click the **Quick Find** icon and make a selection. **Note:** If you map a User Account to two or more NNMi User Groups, NNMi gives the User Account the privileges associated with each mapped NNMi User Group.

Remove a User from a User Group (User Account Mapping)

Only NNMi administrators can add and delete accounts and change NNMi User Accounts and User Groups. See "About User Account Mappings" on page 514.

Tip: NNMi administrators can also use the Security Wizard or command line to complete this task. See "Create and Delete User Accounts Using the Security Wizard" on page 545 or nnmsecurity.ovpl.

To remove a user from an NNMi User Group:

Note: Removing a user from a User Group does not delete the User Account or User Group.

1. Navigate to the **User Account Mappings** view.
   a. From the Workspaces navigation panel, select the **Configuration** workspace.
   b. Expand **Security**.
   c. Select **User Account Mappings**.
2. Select the row representing the User Account mapping you want to change.
3. Delete the User Account mapping by clicking the Delete icon.
4. Click **Save and Close**.

   Note: If you change the User Account mapping for a user who is currently signed into the NNMi console, the change does not take effect until the next time the user signs in. By default, the NNMi timeout limit is 18 hours. If a user has not signed out within 18 hours, NNMi forces the user to sign out. To change the Console Timeout value, see "Configuring the NNMi User Interface" on page 467

Remove User Accounts from User Groups

Tip: NNMi administrators can also use the User Account Mappings view or command line to complete this task. See "Map User Accounts to User Groups (User Account Mapping Form)" on page 553 or nnmsecurity.ovpl.

To remove a User Account mapping from a User Group:

Note: When you remove a User Account from a User Group, you are only deleting the mapping between the two. You are not deleting the User Account or User Group from the NNMi database. See "About User Account Mappings" on page 514 for more information.

1. From the **Security Wizard** main page, select the **Map User Accounts and User Groups** option.
2. Navigate to the **User Account Mapping** table.
3. Select the row that contains the User Account and User Group mapping you want to delete.
4. Click ✖ **Delete**.
5. Repeat steps 3 and 4 to delete each mapping.
6. When you finish, click the **Save and Close** button to save your security configuration:

Note: NNMi does not save any configuration changes until you click **Save and Close** to save your security configuration.

Map User Accounts and User Groups

You can map User Accounts and User Groups using either the Security Wizard main page or using a pop-up dialog box.

- Use the Security Wizard main page:

  "Assign User Accounts to User Groups Using the Security Wizard Page" on page 558

  "Assign User Groups to User Accounts Using the Security Wizard Page" on the next page

- Use the 🗔 pop-up dialog box:

  "Assign User Accounts to User Groups Using the Security Wizard Dialog Box" on page 559

  "Assign User Groups to User Accounts Using the Security Wizard Dialog Box" on the next page

Assign User Groups to User Accounts Using the Security Wizard Page

Tip: You can also use the Security Wizard pop-up dialog box to complete this task. See "Assign User Groups to User Accounts Using the Security Wizard Dialog Box" below for more information.

When using the wizard main page to assign User Groups to User Accounts, note the following (see "About User Account Mappings" on page 514 for more information):

- The **User Account Mapping** table displays the mapping that applies to the selected User Account or User Group.
- Double-click a row or select a row and click to use the **Assign User Groups to User Accounts** dialog box instead of the Wizard main page.
- Your configuration changes are not saved until you click the **Save and Close** button:

  Save & Close

Tip: NNMi administrators can also use the User Account Mappings view or command line to complete this task. See "Map User Accounts to User Groups (User Account Mapping Form)" on page 553 or nnmsecurity.ovpl.

To assign User Groups to User Accounts using the wizard main page:

Tip: To select multiple rows, use Ctrl-Click.

1. From the **Security Wizard** main page, select the **Map User Accounts and User Groups** option.
2. Select a row in the **User Accounts** table.
3. In the **User Groups** table, click the ◁ left arrow in the row of the User Account you want to assign to the selected User Group.

   The selected User Group and User Account names appear in the **User Account Mapping** table.
4. Repeat steps 2 and 3 to assign each User Group you want to the User Account.
5. When you finish, click the **Save and Close** button to save your security configuration:

   Save & Close

   Note: NNMi does not save any configuration changes until you click **Save and Close** to save your security configuration.

Assign User Groups to User Accounts Using the Security Wizard Dialog Box

Tip: You can also use the Security Wizard main page to complete this task. See "Assign User Groups to User Accounts Using the Security Wizard Page" above for more information.

Note the following (see "About User Account Mappings" on page 514 for more information):

- When you select a row in the **User Groups** table, NNMi filters the **User Accounts** table to display only those User Accounts that are not assigned to the selected User Group.
- When you select a row in the **User Accounts** table, NNMi filters the **User Groups** table to display only those User Groups to which the selected User Account has not been assigned.

Tip: NNMi administrators can also use the User Account Mappings view or command line to complete this task. See "Map User Accounts to User Groups (User Account Mapping Form)" on page 553 or nnmsecurity.ovpl.

To assign User Groups to User Accounts using the wizard pop-up dialog box:

Tip: To select multiple rows, use Ctrl-Click.

1. From the **Security Wizard** page, select the **Map User Accounts and User Groups** option.
2. In the **User Accounts** table in the wizard page, double-click the User Account to which you want to assign User Groups or select a row and click to use the **Assign User Groups to User Accounts** dialog instead of the Wizard page.
3. In the wizard dialog box, select a row in the **Available User Groups** table.
4. Click the ▷ right arrow.

   The selected User Group Name appears in the **Assigned to User Groups** table.
5. Repeat steps 3 and 4 to assign each User Group you want to the selected User Account.
6. Click **Close** Close to close the dialog box.
7. When you finish, click the **Save and Close** button to save your security configuration:

   Save & Close

   Note: NNMi does not save any configuration changes until you click **Save and Close** to save your security configuration.

Assign User Accounts to User Groups Using the Security Wizard Page

Tip: You can also use the Security Wizard pop-up dialog box to complete this task. See "Assign User Accounts to User Groups Using the Security Wizard Dialog Box" on the next page for more information.

When using the wizard main page to assign User Accounts to User Groups, note the following (see "About User Account Mappings" on page 514 for more information):

- The **User Account Mapping** table displays the mapping that applies to the selected User Account or User Group.
- Double-click a row or select a row and click to use the **Assign User Accounts to User**

Groups dialog instead of the Wizard page.

- Your configuration changes are not saved until you click **Save and Close**.

For more information about User Accounts, see "About User Account Mappings" on page 514.

> **Tip:** NNMi administrators can also use the User Account Mappings view or command line to complete this task. See "Map User Accounts to User Groups (User Account Mapping Form)" on page 553 or nnmsecurity.ovpl.

To assign User Accounts to User Groups using the wizard main page:

> **Tip:** To select multiple rows, use Ctrl-Click.

1. From the **Security Wizard** page, select the **Map User Accounts and User Groups** option.
2. Select a row in the **User Accounts** table.
3. In the **User Groups** table, click the ⇦ left arrow in the row of the User Group you want to assign to the selected User Account.

   The User Account and User Group names appear in the **User Account Mappings** table.
4. Repeat steps 2 and 3 to assign each User Account you want to a User Group.
5. When you finish, click the **Save and Close** button to save your security configuration:

   > **Note:** NNMi does not save any configuration changes until you click **Save and Close** to save your security configuration.

Assign User Accounts to User Groups Using the Security Wizard Dialog Box

> **Tip:** You can also use the Security Wizard main page to complete this task. See "Assign User Accounts to User Groups Using the Security Wizard Page" on the previous page for more information.

Note the following (see "About User Account Mappings" on page 514 for more information):

- When you select a row in the **User Accounts** table, NNMi filters the **User Groups** table to display only those User Groups to which the selected User Account has not been assigned.
- When you select a row in the **User Groups** table, NNMi filters the **User Accounts** table to display only those User Accounts that are not assigned to the selected User Group.

> **Tip:** NNMi administrators can also use the User Account Mappings view or command line to complete this task. See "Map User Accounts to User Groups (User Account Mapping Form)"

on page 553 or nnmsecurity.ovpl.

To assign User Accounts to User Groups using the wizard pop-up dialog box:

Tip: To select multiple rows, use Ctrl-Click.

1. From the **Security Wizard** main page, select the **Map User Accounts and User Groups** option.
2. In the **User Groups** table in the wizard main page, double-click the User Group to which you want to assign User Accounts or select a row and click to use the **Assign User Accounts to User Groups** dialog instead of the Wizard page.
3. Select a row in the **Available User Accounts** table.
4. Click the ➡ right arrow.

   The selected User Account name appears in the **Assigned to User Accounts** table.
5. Repeat steps 2 through 4 to assign each User Account you want to the User Group.
6. Click **Close** to close the dialog box.
7. In the wizard main page, when you finish your security configuration, click **Save and Close** to save your configuration changes.

Security Group Tasks

NNMi administrators can configure Security Groups to limit node access by using the following methods:

- The Configuration Wizard ("Create and Delete Security Groups Using the Security Wizard" on the next page)
- The Security Accounts view ("Configure Security Groups (Security Group Form)" below)
- The nnmsecurity.ovpl command line tool

Configure Security Groups (Security Group Form)

Required only for Operator or Guest users:

Security Groups enable NNMi administrators to identify groups of nodes that require the same access level. See "About Security Groups" on page 515 for more information.

Use the **Security Groups** form to create, edit, or delete a Security Group.

Tip: NNMi administrators can also use the Security Wizard or command line to complete this task. See "Create and Delete Security Groups Using the Security Wizard" on the next page or nnmsecurity.ovpl.

To configure a Security Group, do the following:

1. Navigate to the **Security Groups** view.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Expand **Security**.
   c. Select **Security Groups**.

   Tip: You can filter the Security Groups table view by User Group.
2. Do one of the following:
   - To create a new configuration, click the ✱ **New** icon.
   - To edit an existing configuration, double-click the Security Groups definition you want to edit.
3. Make your configuration choices. (See the Security Group Attributes table.)
4. Click **Save and Close** to apply your changes.
5. See "Methods for Assigning Nodes to Security Groups" on page 563.

Security Group Attributes

Attribute	Description
Name	Enter the name that uniquely identifies this Security Group. Type a maximum of 255 characters. Alpha-numeric, spaces, and special characters (~ ! @ # $ % ^ & * ( ) _+ -) are permitted.
UUID	NNMi assigns a Universally Unique Object Identifier to the Security Group. This UUID is unique across all databases.
Description	Type a maximum of 2048 characters. Alpha-numeric, spaces, and special characters (~ ! @ # $ % ^ & * ( ) _+ -) are permitted.

Related Topics

"Configure Tenants" on page 194

"About Multi-Tenancy and Global Network Management" on page 95

Create and Delete Security Groups Using the Security Wizard

Required only for Operator or Guest users:

See "About Security Groups" on page 515 for more information.

Tip: NNMi administrators can also use the Security Groups view or command line to complete this task. See "Configure Security Groups (Security Group Form)" on the previous page, or nnmsecurity.ovpl.

To create Security Groups:

1. From the **Security Wizard** main page, do one of the following:

   a. Select the **Map User Groups and Security Groups** option.

   b. Select the **Assign Nodes to Security Groups** option.

2. Navigate to the **Security Groups** table.

3. Click **New**.

4. In the **Create Security Group** dialog box, enter the following:

   a. **Name**: Enter the name that uniquely identifies this Security Group. Type a maximum of 255 characters. Alpha-numeric, spaces, and special characters (~ ! @ # $ % ^ &amp; * ( ) _ + -) are permitted.

   b. **Description**: Type a maximum of 2048 characters. Alpha-numeric, spaces, and special characters (~ ! @ # $ % ^ &amp; * ( ) _+ -) are permitted.

5. Click **Add**.

6. Repeat Step 4 and 5 to add each Security Group.

7. When you finish adding Security Groups, in the **Create Security Group** dialog box, click **Close** .

8. When you finish, click the **Save and Close** button to save your security configuration:

   Note: NNMi does not save any configuration changes until you click **Save and Close** to save your security configuration.

To delete Security Groups:

1. Select a row in the **Security Groups** table.

2. Click **Delete**.

3. When you finish, click the **Save and Close** button to save your security configuration:

   Note: NNMi does not save any configuration changes until you click **Save and Close** to save your security configuration.

Assign Nodes to Security Groups

Required only for Operator or Guest users:

When assigning nodes to Security Groups, note the following (see "About Security Groups" on page 515 for more information):

- When you select a row in the **Security Groups** table, NNMi filters the **Nodes Assigned to Security Group** table to display only those nodes that are assigned to the selected Security Group.

- Your configuration changes are not saved until you click **Save and Close**.

Tip: NNMi administrators can also use other methods to complete this task. See "Methods for Assigning Nodes to Security Groups" below including nnmsecurity.ovpl.

To assign nodes to a Security Group:

1. From the **Security Wizard** main page, select the **Assign Nodes to Security Groups** option.
2. Select a row in the **Security Groups** table.
3. In the **Available Nodes** table, do one of the following:
   a. Select a Node Group in the Node Group filter drop-down list or select a column filter to specify the nodes to be assigned to the Security Group.
   b. User Ctrl-Click to select each node you want to assign to the selected Security Group.
4. Click to specify that you want to assign the selected nodes to the Security Group.

   The **Nodes to be Assigned to Selected Group** table displays the list of nodes to be assigned to the selected Security Group.
5. Repeat steps 2 through 4 to assign nodes to a selected Security Group.
6. When you finish, click the **Save and Close** button to save your security configuration:

   Note: NNMi does not save any configuration changes until you click **Save and Close** to save your security configuration.

Methods for Assigning Nodes to Security Groups

When NNMi discovers nodes in your network environment, Tenant and Security Group settings are established in the following manner:

- **Discovery Seeds**: If Nodes are discovered as Discovery seeds, the NNMi administrator specifies a Tenant for each Discovery Seed. See "Specify Discovery Seeds" on page 256. When NNMi administrators define a Tenant, they specify an **Initial Discovery Security Group**. Any newly discovered Node within the defined Tenant is assigned to this Security Group. NNMi administrators can change either the node's Tenant or Security Group assignment or both at any time.

  Nodes assigned to the *Default Security Group* are visible from all views. To control access to a device, assign that device to a Security Group other than Default Security Group.

  Nodes within one Tenant can each be assigned to different Security Groups, and Nodes within one Security Group each be assigned to different Tenants.

- **Auto-Discovery for Default Tenant**: When you configure Auto-Discovery Rules, NNMi assigns any Nodes discovered using those Auto-Discovery Rules to the *Default Tenant* and whichever Security Group is currently configured as the Default Tenant's Initial Discovery Security Group setting (the *Default Security Group* out-of-box). See "Configure Tenants" on page 194 .

Global Network Management: Regional Managers forward information about Nodes to the Global Manager. The Global Manager's copy of the Node object has the same Tenant assignment as the Regional Manager's record of that Node.

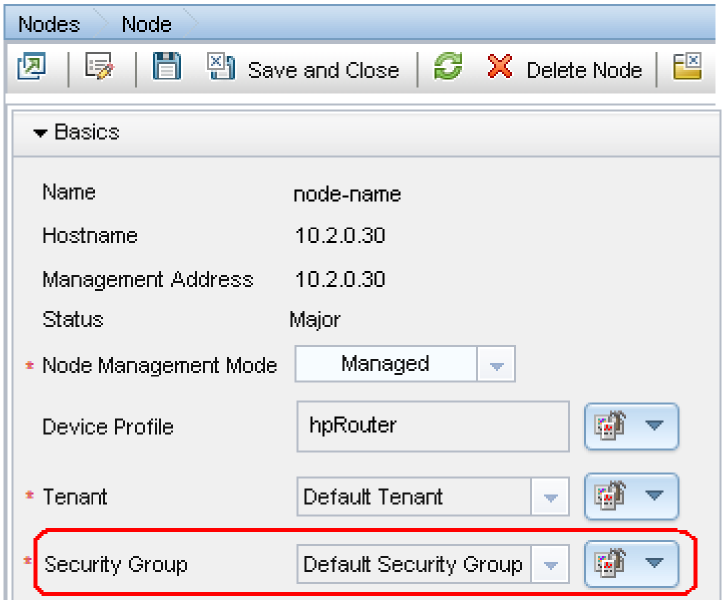
In a Global Network Management environment, best practice is to have the NNMi administrators for the Global Manager and all Regional Managers agree to a predefined list of Tenant names. Those Tenants would be defined on the Regional Managers, the Tenant definitions exported, and those Tenant definitions imported onto the Global Manager (thus ensuring that the UUID and name value for each Tenant match on both NNMi management servers). The NNMi administrator on the Global Manager update their Tenant definitions to assign Initial Discovery Security Group values that make sense for the Global Manager's team. See "About Multi-Tenancy and Global Network Management" on page 95 for more information.

> **Note:** If a Regional Manager forwards information about a Node to the Global Manager, and that Node is assigned to a Tenant object that does not exist on the Global Manager, NNMi creates a Tenant with the UUID and name from the Regional Manager, but creates a new Security Group with that Tenant name (does not duplicate the Regional Manager's setting for that Tenant's *Initial Discovery Security Group* setting). NNMi maps that new Security Group to the following:
>
> - User Group = NNMi Administrator
> - Object Access Privilege = Object Administrator
>
> The Global Manager's NNMi administrator can assign a *different* Initial Discovery Security Group to a Tenant definition at any time. From that point onward, the NNMi Global Manager uses that new Initial Discovery Security Group setting when creating new nodes within that Tenant.

NNMi administrators can change the Security Group assignment for Node objects using the following methods:

- Use the Security Wizard, "Assign Nodes to Security Groups" on page 562 .
- Use the nnmsecurity.ovpl command line tool.
- Use the Node form. However, until an NNMi Administrator defines at least one Security Group in addition to those provided out-of-box by NNMi:
  - The Security Group attribute does not appear on any Node form.

- The Security Group column does not appear in the Nodes (All Attributes) view.



Name node-name
Hostname 10.2.0.30
Management Address 10.2.0.30
Status Major
* Node Management Mode Managed
Device Profile hpRouter
* Tenant Default Tenant
* Security Group Default Security Group

Tip: NNMi administrators can use Security Groups in Node Group definitions that become filters in NNMi views. If an NNMi user cannot access any nodes in a particular Node Group, that filter dynamically disappears from the filter selection list in the NNMi views.

Security Group Mapping Tasks

NNMi administrators can map User Groups to Security Groups using the following methods:

- The Configuration Wizard ("Map User Groups and Security Groups " on page 569)
- The Security Accounts view ("Map User Groups to Security Groups (Security Group Mapping Form)" below)
- The nnmsecurity.ovpl command line tool

Map User Groups to Security Groups (Security Group Mapping Form)

Required only for Operator or Guest users:

See "About Security Group Mappings" on page 516 for more information.

Tip: NNMi administrators can also use the Security Wizard or command line to complete this task. See "Map User Groups and Security Groups " on page 569 and nnmsecurity.ovpl.

To assign a User Group to a Security Group :

1. Navigate to the **Security Group Mappings** view.
   a. From the workspaces navigation panel, select the **Configuration** workspace.

   b. Expand **Security**.

   c. Select **Security Group Mappings**.

   d. Double-click the row representing the Security Group mapping you want to edit.

2. Make your configuration choices. (See the Security Group Mapping Attributes table.)

3. Click **Save and Close** to save your changes and return to the **Security Group Mappings** view.

Security Group Mapping Attributes

Attribute	Description
User Group	Specify the User Group to be assigned to the Security Group. In the **User Group** attribute, click the Lookup icon. • To create new User Group, click the **New** icon and provide the required information. (See "Configure User Groups (User Group Form)" on page 550 for more information.) • To select an NNMi User Group configuration, click the **Quick Find** icon and make a selection.
Security Group	Specify the Security Group to be assigned to the User Group. In the **Security Group** attribute, click the Lookup icon. • To create new Security Group, click the **New** icon and provide the required information. See "Configure Security Groups (Security Group Form)" on page 560 for more information. • To select an Security Group configuration, click the **Quick Find** icon and make a selection.
Object Access Privilege	Determines the level of access each User Account in the User Group has to the nodes assigned to its Security Group. In the **Object Access Privilege** attribute, select a privilege level from the drop-down list. NNMi provides the following privileges: • Object Administrator • Object Operator Level 2 • Object Operator Level 1 (with less access privileges than Level 2) • Object Guest See "Object Access Privileges Provided in NNMi" below for more information.

Object Access Privileges Provided in NNMi

As an NNMi administrator, when you map User Groups to Security Groups, you also determine the Object Access Privilege.

The Object Access Privilege determines the level of access each User Account in the User Group has to the nodes associated with the assigned Security Group. See "Control Menu Access" on page 574 and "Actions Provided by NNMi" on page 43 for more information.

NNMi provides the following Object Access Privileges. Each can be used in any number of Security Group Mappings:

- Object Administrator
- Object Operator Level 2
- Object Operator Level 1 (with less access privileges than Level 2)
- Object Guest

You cannot change the Object Access Privileges definitions that NNMi provides.

For more information about access control, see the following topics:

- "About Security Group Mappings" on page 516
- "Determine which NNMi User Group to Assign" on page 549 (Use to control access to views and forms.)
- "Control Menu Access" on page 574 (NNMi administrators control which roles can access a small subset of Action menu items. The **NNMi Role**[1] is assigned to a User Account through the NNMi User Group.
- "Configure Basic Settings for a Node Group Map" on page 489 (For each Node Group Map, the **Minimum NNMi Role for Saving Map Layout** attribute setting controls the minimum user role required for saving the layout after the user repositions nodes on the map.)

Remove User Groups from Security Group Mappings

Only NNMi administrators can change Security Group mappings. See "About Security Group Mappings" on page 516.

Tip: NNMi administrators can also use the Security Wizard or command line to complete this task. See "Remove User Groups from Security Group Mappings " on page 573or nnmsecurity.ovpl.

To remove a User Group from a Security Group Mapping:

Note: Removing the User Group from a Security Group deletes the mapping between the two (not the User Group or Security Group from the NNMi database).

1. Navigate to the **Security Group Mappings** view.
   a. From the Workspaces navigation panel, select the **Configuration** workspace.
   b. Expand **Security**.
   c. Select **Security Group Mappings**.
2. Select the row representing the Security Group mapping you want to change.

[1] Determined by your membership in one of four special NNMi User Groups (NNMi Administrators, NNMi Level 2 Operators, NNMi Level 1 Operators, or NNMi Guest Users). This membership determines what you can see and do within the NNMi console.

Note: By default, all users assigned to the predefined **NNMi User Group**[1]s see all nodes discovered by NNMi (see "User Groups Provided in NNMi" on page 547). To prevent this, delete the Security Group Mapping for NNMi Level 1 Operators (with less access privileges than Level 2 Operators), NNMi Level 2 Operators, and NNMi Guest. Then, create one or more Security Groups and remap those User Groups to the appropriate Security Group.

3. To delete the Security Group mapping, click the ✖ Delete icon.
4. Click **Save and Close**.

   Note: If you change the Security Group mapping for a user who is currently signed into the NNMi console, the change does not take effect until the next time the user signs in. By default, the NNMi timeout limit is 18 hours. If a user has not signed out within 18 hours, NNMi forces the user to sign out. To change the Console Timeout value, see "Configuring the NNMi User Interface" on page 467.

Change the User Group to Security Group Assignment

Required only for Operator or Guest users:

Only NNMi administrators can change Security Group mappings. See "About Security Group Mappings" on page 516.

Tip: NNMi administrators can also use the Security Wizard or command line to complete this task. See "Remove User Groups from Security Group Mappings " on page 573or nnmsecurity.ovpl.

To change the User Group to Security Groups assignment use the following instructions:

Note: To change a User Group to Security Group assignment, you first delete the existing Security Group mapping.

1. Navigate to the **Security Group Mappings** view.
   a. From the workspaces navigation panel, select the **Configuration** workspace.
   b. Expand **Security**.
   c. Select **Security Group Mappings**.
2. Select the row representing the Security Group mapping you want to change.
3. Delete the Security Group mapping by clicking the ✖ Delete icon.
4. Select the ✳ New icon to configure the new Security Group mapping.
5. Make your configuration choices. (See the Security Group Mapping Attributes table.)
6. Click **Save and Close** to save your changes and return to the **Security Group Mappings** view.

[1] NNMi User Groups are those User Groups provided by NNMi. Users cannot access the NNMi console until their User Account is mapped to at least one of the following NNMi User Groups: NNMi Administrators, NNMi Level 2 Operators, NNMi Level 1 Operators (with less access privileges than Level 2 Operators), and NNMi Guest Users

Note: If you change the User Group to Security Group mapping for a user who is currently signed into the NNMi console, the change does not take effect until the next time the user signs in. By default, the NNMi timeout limit is 18 hours. If a user has not signed out within 18 hours, NNMi forces the user to sign out. To change the Console Timeout value, see "Configuring the NNMi User Interface" on page 467

Security Group Mapping Attributes

Attribute	Description
User Group	Specify the User Group to be assigned to the Security Group. In the **User Group** attribute, click the Lookup icon. • To create new User Group, click the ✱ **New** icon and provide the required information. (See "Configure User Groups (User Group Form)" on page 550 for more information.) • To select an User Group configuration, click the **Quick Find** icon and make a selection.
Security Group	Specify the Security Group to be assigned to the User Group. In the **Security Group** attribute, click the Lookup icon. • To create new Security Group, click the ✱ **New** icon and provide the required information. See "Configure Security Groups (Security Group Form)" on page 560 for more information. • To select an NNMi Security Group configuration, click the **Quick Find** icon and make a selection.
Object Access Privilege	Determines the level of access each User Account in the User Group has to the nodes assigned to its Security Group. In the **Object Access Privilege** attribute, select a privilege from the drop-down list. NNMi provides the following privileges: • Object Administrator • Object Operator Level 2 • Object Operator Level 1 (with less access privileges than Level 2) • Object Guest See "Object Access Privileges Provided in NNMi" on page 566 for more information.

Map User Groups and Security Groups

Required only for Operator or Guest users:

You can map User Groups and Security Groups using either the Security Wizard main page or using a pop-up dialog box.

- Use the Security Wizard main page:

  "Assign User Groups to Security Groups Using the Security Wizard Page" on the next page

  "Assign Security Groups to User Groups Using the Security Wizard Page" below

- Use the pop-up dialog box:

  "Assign User Groups to Security Groups Using the Security Wizard Dialog Box" on page 572

  "Assign Security Groups to User Groups Using the Security Wizard Dialog Box" on the next page

Assign Security Groups to User Groups Using the Security Wizard Page

Required only for Operator or Guest users:

Tip: You can also use the Security Wizard pop-up dialog box to complete this task. See "Assign Security Groups to User Groups Using the Security Wizard Dialog Box" on the next page for more information.

When using the wizard main page to assign Security Groups to User Groups, note the following (see "About Security Group Mappings" on page 516 for more information):

- The **Security Group Mapping** table displays the mapping that applies to the selected User Group or Security Group.
- Double-click a row or select a row and click to use the **Assign Security Groups to User Groups** dialog instead of the Wizard page.
- Your configuration changes are not saved until you click the **Save and Close** button:

Tip: NNMi administrators can also use the Security Group Mappings view or command line to complete this task. See "Map User Groups to Security Groups (Security Group Mapping Form)" on page 565 or nnmsecurity.ovpl.

To assign Security Groups to User Groups using the wizard main page:

Tip: To select multiple rows, use Ctrl-Click.

1. From the **Security Wizard** page, select the **Map User Groups and Security Groups** option.
2. Select a row in the **Security Groups** table.
3. In the **User Groups** table, click the ⇨ right arrow in the row of the User Group you want to assign to the selected Security Group.

   The Security Group and User Group names appear in the **Security Group Mapping** table.
4. Repeat steps 2 and 3 to assign each Security Group you want to a User Group.
5. When you finish, click the **Save and Close** button to save your security configuration:

Save & Close

Note: NNMi does not save any configuration changes until you click **Save and Close** to save your security configuration.

Assign Security Groups to User Groups Using the Security Wizard Dialog Box

Required only for Operator or Guest users:

Tip: You can also use the Security Wizard main page to complete this task. See "Assign Security Groups to User Groups Using the Security Wizard Page" on the previous page for more information.

See "About Security Group Mappings" on page 516 for more information.

Tip: NNMi administrators can also use the Security Group Mappings view or command line to complete this task. See "Map User Groups to Security Groups (Security Group Mapping Form)" on page 565 or nnmsecurity.ovpl.

To assign Security Groups to User Groups using the wizard pop-up dialog box:

Tip: To select multiple rows, use Ctrl-Click.

1. From the **Security Wizard** page, select the **Map User Groups and Security Groups** option.
2. In the **User Groups** table in the wizard main page, double-click the User Group to which you want to assign Security Groups or select a row and click to use the **Assign Security Groups to User Groups** dialog instead of the Wizard page.
3. In the wizard dialog box, select a row in the **Available Security Groups** table.
4. Click the ⇨ right arrow.

   The selected Security Group Name appears in the **Assigned to Security Groups** table.
5. Repeat steps 2 through 4 to assign each Security Group you want to the User Group.
6. Click **Close** to close the dialog box.
7. When you finish, click the **Save and Close** button to save your security configuration:

   Save & Close

   Note: NNMi does not save any configuration changes until you click **Save and Close** to save your security configuration.

Assign User Groups to Security Groups Using the Security Wizard Page

Required only for Operator or Guest users:

Tip: You can also use the Security Wizard pop-up dialog box to complete this task. See "Assign User Groups to Security Groups Using the Security Wizard Dialog Box" below for more information.

When using the wizard main page to assign User Groups to Security Groups, note the following (see "About Security Group Mappings" on page 516 for more information):

- The **Security Group Mapping** table displays the mapping that applies to the selected User Group or Security Group.
- Double-click a row or select a row and click to use the **Assign User Groups to Security Groups** dialog instead of the wizard main page.
- Your configuration changes are not saved until you click the **Save and Close** button:

Save & Close

Tip: NNMi administrators can also use the Security Group Mappings view or command line to complete this task. See "Map User Groups to Security Groups (Security Group Mapping Form)" on page 565 or nnmsecurity.ovpl.

To assign User Groups to Security Groups using the wizard main page:

Tip: To select multiple rows, use Ctrl-Click.

1. From the **Security Wizard** main page, select the **Map User Groups and Security Groups** option.
2. Select a row in the **User Groups** table.
3. In the **Security Groups** table, select the ⇦ left arrow in the row of the Security Group you want to assign to the selected User Group.

   The User Group and Security Group names appear in the **Security Group Mapping** table.
4. Repeat steps 2 and 3 to assign each User Account you want to a User Group.
5. When you finish, click the **Save and Close** button to save your security configuration:

   Save & Close

   Note: NNMi does not save any configuration changes until you click **Save and Close** to save your security configuration.

Assign User Groups to Security Groups Using the Security Wizard Dialog Box

Required only for Operator or Guest users:

Tip: You can also use the main Security Wizard page to complete this task. See "Assign User Groups to Security Groups Using the Security Wizard Page" on the previous page for more information.

See "About Security Group Mappings" on page 516 for more information.

Tip: NNMi administrators can also use the Security Group Mappings view or command line to complete this task. See "Map User Groups to Security Groups (Security Group Mapping Form)" on page 565 or nnmsecurity.ovpl.

To assign User Groups to Security Groups using the wizard pop-up dialog box:

Tip: To select multiple rows, use Ctrl-Click.

1. From the **Security Wizard** main page, select the **Map User Groups and Security Groups** option.
2. In the **Security Groups** table in the wizard page, double-click the Security Group to which you want to assign User Groups or select a row and click ▤ to use the **Assign User Groups to Security Groups** dialog instead of the Wizard page.
3. In the wizard dialog box, select a row in the **Available User Groups** table.
4. Click the ⇨ right arrow.

   The selected User Group Name appears in the **Assigned to User Groups** table.
5. Repeat steps 3 and 4 to assign each User Group you want to the Security Group.
6. Click **Close** to close the dialog box.
7. When you finish, click the **Save and Close** button to save your security configuration:

   Save & Close

   Note: NNMi does not save any configuration changes until you click **Save and Close** to save your security configuration.

Remove User Groups from Security Group Mappings

Tip: NNMi administrators can also use the Security Group Mappings view or the command line to complete this task. See "Remove User Groups from Security Group Mappings" on page 567 or nnmsecurity.ovpl.

When the NNMi administrator removes a User Group from a Security Group Mapping, NNMi only deletes the mapping between the two (not the User Group or Security Group from the NNMi database). See "About Security Groups" on page 515 for more information.

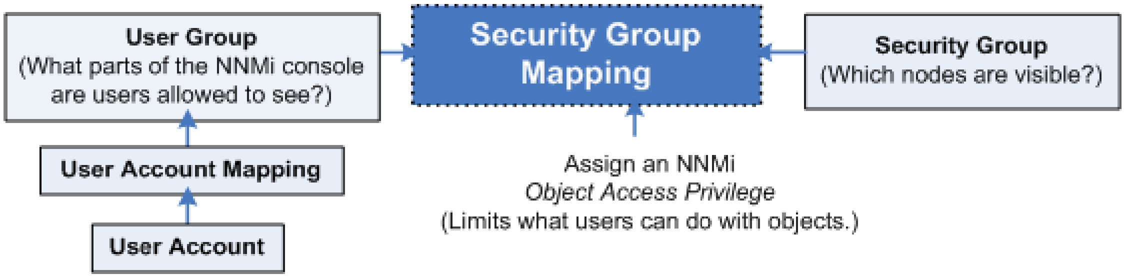
To remove a User Group from a Security Group Mapping:

1. From the **Security Wizard** main page, select the **Map User Groups and Security Groups** option.
2. Navigate to the **Security Group Mapping** table.
3. Select the row that contains the User Group and Security Group mapping you want to delete.
4. Click ✖ **Delete**.
5. Repeat steps 3 and 4 to delete each mapping.
6. When you finish, click the **Save and Close** button to save your security configuration:

Note: NNMi does not save any configuration changes until you click **Save and Close** to save your security configuration.

Control Menu Access

Access to the Tools and Actions menu items is controlled by Security Group Mapping configuration settings: User Group, Security Group, and *Object Access Privilege*


User Group
(What parts of the NNMi console are users allowed to see?)
Security Group Mapping
Security Group
(Which nodes are visible?)
User Account Mapping
User Account
Assign an NNMi
Object Access Privilege
(Limits what users can do with objects.)

See "Determine which NNMi User Group to Assign" on page 549 for additional information about User Group limitations. See "Object Access Privileges Provided in NNMi" on page 566 and "Actions Provided by NNMi" on page 43 for additional information about *Object Access Privileges*.

Tip: You can right-click any object in a table or map view to access the **Actions** menu.

Note the following:

- User Groups determine access to NNMi console workspaces, views and forms. User Groups also determine the Tools and Actions that the users in the User Group can access.
- You MUST assign each User Account to one of the predefined **NNMi User Group**[1]s before that user can access NNMi. See "User Groups Provided in NNMi" on page 547 for more information.
- If you map a User Account to two or more NNMi User Groups, NNMi gives the User Account the privileges associated with each User Group to which the User Account is assigned.
- Security Groups are optional and control (through User Groups) which Users can access a node and its hosted objects, such as an interface. Each node is associated with only one Security Group.

  Note:Users see only those members of an object group (for example, Node Group or Router Redundancy Group) for which they have access. If a user cannot access any nodes in the group, the group is not visible to that user.
- Object Access Privileges are associated only with Security Groups and their associated User Groups. Object Access Privileges determine the Tools and Actions that the User Group can

[1] NNMi User Groups are those User Groups provided by NNMi. Users cannot access the NNMi console until their User Account is mapped to at least one of the following NNMi User Groups: NNMi Administrators, NNMi Level 2 Operators, NNMi Level 1 Operators (with less access privileges than Level 2 Operators), and NNMi Guest Users

access for the nodes they are permitted to view.

- If a User Account is assigned an NNMi User Group with *more privileges* than the Object Access Privilege, the user sees all of the actions available for the User Group (not restricted because of the Object Access Privilege setting). For example, if a User Account is assigned to the User Group **NNMi Level 2 Operators** and has an Object Access Privilege of **Object Operator Level 1** (with less access privileges than Level 2 Operators) for a set of nodes, the operator sees all actions available to Level 2 Operators.
- If a User Account is assigned an NNMi User Group with *less privileges* than the Object Access Privilege, the user will not see all of the actions available for the Object Access Privilege. For example, if a User Account is assigned to the User Group **NNMi Level 1 Operators** (with less access privileges than Level 2 Operators) and has an Object Access Privilege of **Object Operator Level 2** for a set of nodes, the operator will see only those actions available to Level 1 Operators. As an NNMi administrator, you must do either of the following:
    - Configure the **Menu Item Context Basic Details** to change the **Required NNMi Role** for the menu item
    - Assign the operator User Account to the **NNMi Level 2 Operators** User Group.

- All menu items are visible to users, but an *Access Denied* message displays when any user with insufficient privileges tries to use a menu item. For example, both Level 1 or Level 2 Operators are denied access to the Communication Settings action.
- You can restrict access to certain Launch Actions (provide tighter security than those enforced by the default settings). See "Configure Menu Item Context Basic Details" on page 1420 for more information about configuring actions.
- If the menu item does not require node access, (for example, **Status Details** for a Node Group) NNMi uses the privileges assigned to the NNMi User Group that is mapped to the User Account.

User Group and Object Access Privilege Required for the Tools Menu:

Access to the NNMi Tools menu items is determined by User Group and the Security Group Object Access Privilege that is set for the node. Also see "Actions Provided by NNMi" on page 43. Click here for information about Tools Menu Access Limitations.

NNMi Tools Menu Access Limitation

Tools Menu Item	NNMi User Group	Object Access Privilege
Find Node	NNMi Guest Users	Object Guest
Find Attached Switch Port	NNMi Level 2 Operators	Object Operator Level 2
Incident Actions Log	NNMi Administrators	Object Administrator
Load /Unload MIB	NNMi Administrators	Object Administrator
MIB Browser	NNMi Level 2 Operators	Object Operator Level 2
NNMi Self-Monitoring Graphs	NNMi Administrators	Object Administrator

NNMi Tools Menu Access Limitation, continued

Tools Menu Item	NNMi User Group	Object Access Privilege
NNMi Status	NNMi Level 1 Operators	Object Operator Level 1
Restore All Default View Settings	NNMi Guest Users	Object Guest
Security Reports	NNMi Administrators	Object Administrator
Signed In Users	NNMi Administrators	Object Administrator
Sign In/Sign Out Audit Log	NNMi Administrators	Object Administrator
Status Distribution Graphs	NNMi Level 2 Operators	Object Operator Level 2
Trap Analytics (iSPI NET only)	NNMi Administrators	Object Administrator
Upload Local MIB File	NNMi Administrators	Object Administrator
Visio Export (iSPI NET only)	NNMi Level 2 Operators	Object Operator Level 2

User Group and Object Access Privilege Required for the Actions Menu:

Access to the NNMi Actions menu is determined by User Group and the Security Group Object Access Privilege that is set for the node. Click here for more information about URL Action Access Limitations.

URL Action Access Limitations

Action Menu Item	Submenu Item	NNMi User Group	Object Access Privilege
Configuration Details	Communication Settings	NNMi Administrators	Object Administrator
Configuration Details	Monitoring Settings	NNMi Level 1 Operators	Object Operator Level1
Custom Attributes		NNMi Administrators	Object Administrator
Graphs		NNMi Level 1 Operators	Object Operator Level 1
Management Mode		NNMi Level 2 Operators	Object Operator Level 2
MIB Information	Browse MIB	NNMi Level 2 Operators	Object Operator Level 2
MIB Information	List Supported MIBs	NNMi Level 2 Operators	Object Operator Level 2
Node Access	Ping (from server)	NNMi Level 1 Operators	Object Operator Level 1

URL Action Access Limitations, continued

Action Menu Item	Submenu Item	NNMi User Group	Object Access Privilege
Node Access	Secure Shell (from client)	NNMi Level 2 Operators	Object Operator Level 2
Node Access	Traceroute (from server)	NNMi Level 1 Operators	Object Operator Level 1
Node Access	Telnet...(from client)	NNMi Level 2 Operators	Object Operator Level 2
Node Group Details	Show All Incidents	NNMi Level 1 Operators	Object Operator Level 1
Node Group Details	Show Members (Include Child Groups)	NNMi Level 1 Operators	Object Operator Level 1
Node Group Details	Preview Members (Current Group Only)	NNMi Level 1 Operators	Object Operator Level 1
Node Group Details	Status Details	NNMi Level 1 Operators	Object Operator Level 1
Node Group Details	Show All Open Incidents	NNMi Level 1 Operators	Object Operator Level 1
Node Group Membership		NNMi Administrators	Object Administrator
Polling	Configuration Poll	NNMi Level 2 Operators	Object Operator Level 2
Polling	Status Poll	NNMi Level 2 Operators	Object Operator Level 2
Show Attached End Nodes		NNMi Guest Users	Object Guest

See Investigate and Diagnose Network Problems for more information about these actions.

Note: Each Tools and Action menu item provided by NNMi is also associated with a *default NNMi Role*. (To determine the *default NNMi Role* assigned to each Action menu item, see "Actions Provided by NNMi" on page 43.) If you change the setting for a Menu Item provided by NNMi to a Role that is a *lower level Role* than the *default NNMi Role* assigned to the menu item, NNMi ignores that change. Any User Group with the lower level Role than the *default NNMi Role* cannot access the menu item.

Set Up Command Line Access to NNMi

NNMi limits access to Command Line Interface (CLI) commands in one of two ways:

- Method One: Requiring User Name and Password.
- Method Two: Requiring permission to access NNMi as the `system` user.

See **Help** → **Documentation Library** → **Reference Pages** for a list of command line commands. Check the appropriate Reference Page to determine which method applies.

Method One: Requiring User Name and Password.

There are two strategies for CLI user name and password:

- Providing the appropriate NNMi User Name attribute value and NNMi Password attribute value within the CLI syntax (`-u` and `-p`).
- Configuring a valid NNMi User Name attribute value and NNMi Password attribute value using the `nnmsetcmduserpw.ovpl` command. See **Help** → **Documentation Library** → **Reference Pages** for details.

  > **Note:** With `nnmsetcmduserpw.ovpl`, the CLI command must then be run on the same machine where the `nnmsetcmduserpw.ovpl` command was executed.

Method Two: Requiring permission to access NNMi as the `system` user.

During NNMi installation, the first access to the NNMi console requires a special `system` User Name and Password. Thereafter, only the following situations are appropriate for the `system` user:

- The CLI you are using runs only when executed by the special NNMi `system` user.
- If your network environment uses X.509 Certificates such as Public Key Infrastructure (PKI) user authentication, all NNMi CLI commands must be executed by the special NNMi `system` user. See the “Configuring NNMi to Support Public Key Infrastructure User Authentication” chapter in the*HP Network Node Manager i Software Deployment Reference*for more information, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.
- For Troubleshooting purposes when mistakes were made that result in zero NNMi users being mapped to the **NNMi User Groups**[1]: *NNMi Administrators*. For more information, see "Restore the Administrator NNMi Role" on page 587.

If method two is required, your CLI command must be issued from the NNMi management server and you must have *read* access to the following file on the NNMi management server: `nms-users.properties`

> **Caution:** Any user with `read` access to the `nms-users.properties` file can potentially sign into the NNMi console and perform Administrator operations.

Note the following for Public Key Infrastructure (PKI) user authentication to provide NNMi User Name and NNMi Password:

[1] NNMi User Groups are those User Groups provided by NNMi. Users cannot access the NNMi console until their User Account is mapped to at least one of the following NNMi User Groups: NNMi Administrators, NNMi Level 2 Operators, NNMi Level 1 Operators (with less access privileges than Level 2 Operators), and NNMi Guest Users

- If you are logged into the operating system as `root` user, NNMi automatically accesses the `system` User Account and runs the command using the NNMi `system` user's credentials.
- If you are logged into the operating system with a user name other than `root` and your user name is not configured for *read* access to the `nms-users.properties` file, NNMi cannot run theCLI command.

Communicate Console Access Information to Your Team

After configuring user passwords and roles, communicate the following information to your team:

- "Open the NNMi Console" below
- "Configuring Sign-In to the NNMi Console" on page 581
- "Sign Out from the Console" on page 581

Open the NNMi Console

Provide each user with the following information:

`http://<serverName>:<portNumber>/nnm/`

> **Note:** If the NNMi Web server uses the https protocol, use `https` instead of `http`. See the "Working with Certificates for NNMi" chapter in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.

`<serverName>` = the fully-qualified domain name of the NNMi management server (values allowed here are determined by the *Enable URL Redirect* setting in User Interface Configuration, see "Configuring the NNMi User Interface" on page 467)

`<portNumber>` = the NNMi HTTP port number

When your NNMi management server has more than one fully-qualified domain name, NNMi chooses one during the installation process. There are two ways to find out which domain name NNMi is using in your network environment:

- Click **Help** → **System Information** and navigate to the **Server** tab. Locate the **Official Fully Qualified Domain Name (FQDN)** attribute value.
- Use the `nnmofficialfqdn.ovpl` command. See the nnmofficialfqdn.ovpl Reference Page.

To determine the current port number configuration, look at the line for `boss.http.port` in the `nms-local.properties` file (see table for the location of this file). See the nnm.ports Reference Page for more information.

Determine the NNMi Console Port Number

Operating System	Identify Current Port Number
Windows	`%NnmDataDir%\conf\nnm\props\nms-local.properties`
UNIX	`/var/opt/OV/conf/nnm/props/nms-local.properties`

Communicate the following browser requirements for your team to use the NNMi console:

- Pop-ups, cookies, and JavaScript must be enabled.
- Each user's screen resolution must be 1024x768 pixels or higher.
- When using Microsoft Internet Explorer as your browser, you can access multiple browser sessions of NNMi. Use a different user name for each browser session.
- When using Mozilla Firefox as your browser, multiple browser sessions all point to the same window.

> **Note:** Users can bookmark the URL for the NNMi console. Use the URL for the NNMi console rather than the NNMi Welcome page. See About the NNMi Console for more information about the NNMi console.

To open the console:

1. Type the following URL (Uniform Resource Locator) into your browser navigation bar:

   `http://<serverName>:<portNumber>/nnm/`

2. Sign in with the following name and password:

   <*name you configured*>

   <*password you configured*>

   > **Tip:** Tip: You can include name and password in the URL. See "Launch the Console (showMain)" on page 1494

3. Click the **Sign In** button. (See "Configuring Sign-In to the NNMi Console" on the next page if you need more information.)
4. The console opens in a new window.
5. *Optional*. Close the NNMi Welcome page.

   > **Note:** If you do not close the NNMi Welcome page or sign out, you can relaunch the console from the NNMi Welcome Page without signing in again.

To refresh the console window:

Click the ⟳ Refresh icon in the tool bar of any NNMi window.

Configuring Sign-In to the NNMi Console

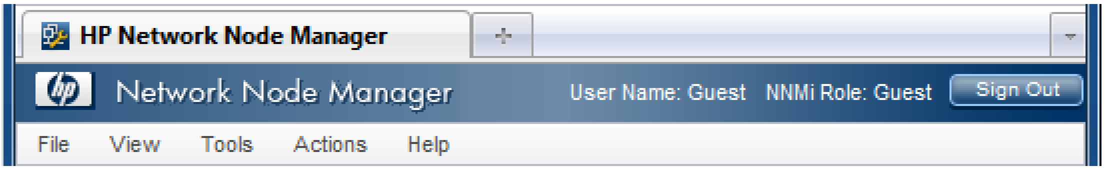
After entering the URL to access the NNMi console (provided by your NNMi administrator), one of the following happens:

- NNMi prompts you to sign into the console:
  a. At the **User Name** prompt, enter the user name that was provided by your NNMi administrator.
  b. At the **Password** prompt, enter the password that was provided by your NNMi administrator.
  c. Click the **Sign In** button.
- If your network environment uses X509 Certificates such as Public Key Infrastructure (PKI) user authentication, the NNMi console opens immediately without requesting a User Name or Password.

> **Note:** The NNMi administrator must configure NNMi to acknowledge your network environment's Public Key Infrastructure (PKI) setup. The PKI configuration maps certificates to NNMi User Accounts. After PKI is configured, NNMi reads the PKI certificate to obtain the NNMi user name information. Steps required to sign in to the NNMi console with certificate validation depend on the user environment. Be sure to communicate these requirements to your team. The NNMi administrator must still define User Accounts within NNMi or configure NNMi to use Lightweight Directory Access Protocol (LDAP), see "X.509 Certificates to Control NNMi Access" on page 507. For more information about PKI configuration, see “Configuring NNMi to Support Public Key Infrastructure User Authentication” chapter in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals.`.

(*NNMi Advanced*) Single Sign-On (SSO) can be configured to enable access to an NNMi Regional Manager through the NNMi Global Manager. For more information about SSO, see “Configuring Single Sign-On for Global Network Management” section in the *HP Network Node Manager i Software Deployment Reference*.

After a user accesses the NNMi console, the User Account name and the highest associated object access privilege appear in the upper right corner of the console as shown in the example below:



Sign Out from the Console

To sign out from the console:

1. Select **File** → **Sign Out**.
2. Click **OK**.

Note the following:

- Sign in is not preserved across user sessions. After signing out, each user must sign in again.
- You must sign out of each browser session that is running NNMi. For example, if you have signed in twice with two different browsers, signing out in one browser does not cause you to lose access in the other browser.
- By default, NNMi automatically signs out any user after 18 hours of inactivity. The NNMi administrator can configure this amount of time. See "Configuring the NNMi User Interface" on page 467.

Troubleshoot NNMi Access

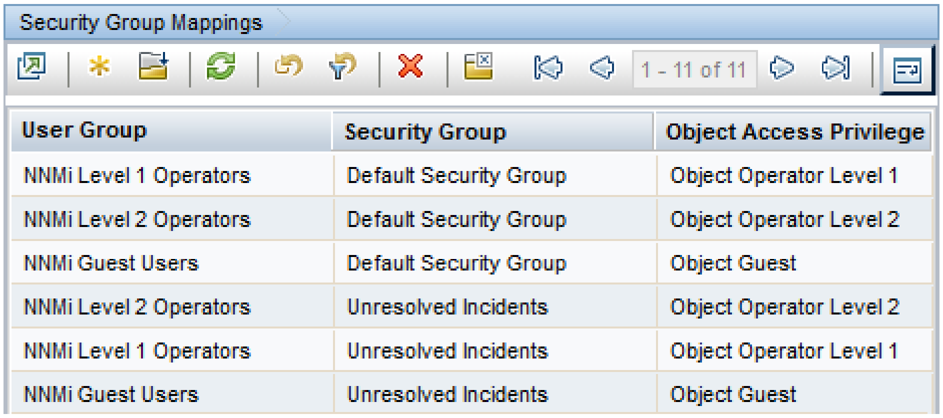
> **Tip:** Select **Help** → **System Information** to view the User Name, NNMi Role, and User Group for the current NNMi session.

NNMi provides several tools to help you troubleshoot and monitor NNMi access:

- "Check Security Configuration" on page 584
- "View the Users who are Signed In to NNMi" on page 585
- "Audit NNMi User Activity" on page 586
- "Restore the Administrator NNMi Role" on page 587
- "Restore NNMi Access for the system User" on page 588

Out-of-box, NNMi Security works in the following manner:

- NNMi assigns all nodes to the Default Security Group.
- NNMi operators and guests can see all discovered nodes and all incidents, because of the default Security Group Mappings:



User Group	Security Group	Object Access Privilege
NNMi Level 1 Operators	Default Security Group	Object Operator Level 1
NNMi Level 2 Operators	Default Security Group	Object Operator Level 2
NNMi Guest Users	Default Security Group	Object Guest
NNMi Level 2 Operators	Unresolved Incidents	Object Operator Level 2
NNMi Level 1 Operators	Unresolved Incidents	Object Operator Level 1
NNMi Guest Users	Unresolved Incidents	Object Guest

Tip: NNMi administrators always see all nodes and incidents, no Security Group Mappings are required for NNMi administrators.

NNMi administrators can limit access to nodes and incidents by deleting the default (out-of-box) Security Group Mappings. Then no operators or guests can access any nodes until an NNMi administrator explicitly adds new, more restrictive Security Group Mappings. When these out-of-box Security Group Mappings are removed, the predefined **NNMi User Group**[1]s provide access to the NNMi console only, rather than to the NNMi console and to all nodes. See "Remove User Groups from Security Group Mappings" on page 567 for more information.

Security Group Mappings have three components:

- User Group identifies the *NNMi users*.
- Security Group identifies *a set of nodes* (and indirectly their hosted objects).
- *Object Access Privilege* determines the level of access that each User Account in the User Group has to the nodes in the associated Security Group.

Each node is associated with one and only one Security Group. NNMi operators and guests can view a node only if one of the User Groups to which that NNMi user belongs is associated with that node's Security Group.

When NNMi discovers nodes in your network environment, Tenant and Security Group settings are established in the following manner:

- **Discovery Seeds**: If Nodes are discovered as Discovery seeds, the NNMi administrator specifies a Tenant for each Discovery Seed. See "Specify Discovery Seeds" on page 256. When NNMi administrators define a Tenant, they specify an **Initial Discovery Security Group**. Any newly discovered Node within the defined Tenant is assigned to this Security Group. NNMi administrators can change either the node's Tenant or Security Group assignment or both at any time.

  Nodes assigned to the *Default Security Group* are visible from all views. To control access to a device, assign that device to a Security Group other than Default Security Group.

  Nodes within one Tenant can each be assigned to different Security Groups, and Nodes within one Security Group each be assigned to different Tenants.

- **Auto-Discovery for Default Tenant**: When you configure Auto-Discovery Rules, NNMi assigns any Nodes discovered using those Auto-Discovery Rules to the *Default Tenant* and whichever Security Group is currently configured as the Default Tenant's Initial Discovery Security Group setting (the *Default Security Group* out-of-box). See "Configure Tenants" on page 194 .

Global Network Management: Regional Managers forward information about Nodes to the Global Manager. The Global Manager's copy of the Node object has the same Tenant assignment as the Regional Manager's record of that Node.

In a Global Network Management environment, best practice is to have the NNMi administrators for the Global Manager and all Regional Managers agree to a predefined list of Tenant names. Those Tenants would be defined on the Regional Managers, the Tenant definitions exported, and those Tenant definitions imported onto the Global Manager (thus ensuring that the UUID and name value for each Tenant match on both NNMi management servers). The NNMi administrator on the Global Manager update their Tenant definitions to assign Initial Discovery Security Group values that make sense for the Global Manager's team. See "About Multi-Tenancy and Global Network Management" on page 95 for more information.

[1] NNMi User Groups are those User Groups provided by NNMi. Users cannot access the NNMi console until their User Account is mapped to at least one of the following NNMi User Groups: NNMi Administrators, NNMi Level 2 Operators, NNMi Level 1 Operators (with less access privileges than Level 2 Operators), and NNMi Guest Users

> **Note:** If a Regional Manager forwards information about a Node to the Global Manager, and that Node is assigned to a Tenant object that does not exist on the Global Manager, NNMi creates a Tenant with the UUID and name from the Regional Manager, but creates a new Security Group with that Tenant name (does not duplicate the Regional Manager's setting for that Tenant's *Initial Discovery Security Group* setting). NNMi maps that new Security Group to the following:
>
> - User Group = NNMi Administrator
> - Object Access Privilege = Object Administrator
>
> The Global Manager's NNMi administrator can assign a *different* Initial Discovery Security Group to a Tenant definition at any time. From that point onward, the NNMi Global Manager uses that new Initial Discovery Security Group setting when creating new nodes within that Tenant.

Node revisions: NNMi administrators can change the Node's initial Security Group assignment. See "Methods for Assigning Nodes to Security Groups" on page 563.

Tip: NNMi administrators can use Security Groups in Node Group definitions that become filters in NNMi views. If a user cannot access any nodes in a particular Node Group, that filter dynamically disappears from the filter selection list in the user's NNMi views. See "Specify Node Group Additional Filters" on page 298 for more information about Node Group filters.

Security influences incidents:

- Network operators and guests can view incidents associated with a node only if that user's User Account is mapped to one of the User Groups that are mapped to the node's Security Group. See "About Security Groups" on page 515 and "About Security Group Mappings" on page 516.
- Any incident that does not have an associated node is assigned to the **Unresolved Incidents** Security Group and NNMi's out-of-box configuration makes these incidents visible to all User Groups. Examples of incidents that are unresolved include unresolved traps, system health, and license violation incidents.
- Operators should only be assigned incidents for nodes they can access.

Check Security Configuration

Each NNMi user can be assigned to multiple Security Group Mappings. The *Object Access Privilege* determines what NNMi users can do with a node object. For example, if their User Group is **NNMi Level 2 Operators**, but the Object Access Privilege is **Object Operator Level 1** (with less access privileges than Level 2), each user assigned to the Security Group Mapping *sees* all of the actions available to a Level 2 Operator, but can run only those *actions allowed* for Level 1 Operators. If an NNMi user is assigned to multiple Security Group Mappings, that user sees all the parts of NNMi that are provided to the highest User Group setting and access for each node is determined by the node's Security Group Mapping.

NNMi administrators can generate a report of possible Security configuration problems:

- Users Accounts that are not mapped to a User Group
- User Accounts that are not mapped to an NNMi User Group

- User Accounts that have unusual NNMi role combinations
- Security Groups that include nodes from multiple tenants
- Empty User Groups and Security Groups
- Tenants with the same name
- Security Groups with the same name

Generate the report using any of the following methods:

- **Tools** → **Security Report**
- The nnmsecurity.ovpl command

You can also use the View Summary of Changes option in the Security Wizard to view a report based on only your latest configuration changes.

View Summary of Changes in the Security Wizard

Use the Security Wizard **View Summary of Changes** option to view your recent configuration changes, including the following:

- The User Accounts created.
- The User Groups created.
- The Security Groups created.
- The User Accounts and User Groups mappings.
- The User Groups and Security Groups mappings.
- The Security Groups that have new nodes assigned to them.

To view the summary of security configuration changes:

From the **Security Wizard** main page, select the **View Summary of Changes** option.

NNMi displays a summary of the configuration changes made since you last saved your changes.

View the Users who are Signed In to NNMi

Use the **Tools** → **Signed in Users** menu option to view a list of the NNMi users who are currently signed in to NNMi. This tool is useful when you want to determine which users and systems are available. For example, you might want to view the users who are signed in before shutting down a system.

To see the list of users who are currently signed in to NNMi:

Select **Tools** → **Signed In Users**.

NNMi displays the number of users currently signed in to NNMi as well as each user name, IP address of the client that is running the NNMi console, and the time in which the user signed in to NNMi.

Audit NNMi User Activity

NNMi tracks a history of sign-in and sign-out activity for each NNMi user. This auditing information also includes a variety of information about user activity since the NNMi management server was last restarted.

NNMi stores the audit log files in the following directory:

- **Windows**:
  `%NnmDataDir%\log\nnm\`
- **UNIX**:
  `/var/opt/OV/log/nnm/`

NNMi stores these log files in a name.log file name format. Any archived log file has a number appended to it in the form `.name.log.%g`.

- *name* is the log file base name
- *%g* represents the archive number of the archived log file

The highest appended archive number represents the oldest file. A log file can become an archived log file after the size of the log file exceeds the configured limit. After a log file exceeds the configured limit, the last active log file is archived. For example, after NNMi archives the `nnm.log` file as the `nnm.log.1` file, NNMi begins logging to a new `nnm.log` file.

To see the most recent sign-in audit report:

1. A tool is available to NNMi administrators. In the console menu bar, select **Tools** → **Sign In/Out Audit Log**.

   Note: If you do not see the **Tools** > **Sign In/Out Audit Log** option, you must enable the log file.

2. The log provides a variety of information about recent account activity. For example:

   ```
   Sign In/Sign Out Audit Log
   Jun 14, 2007 10:53:01.926 AM [ThreadID:719]
   com.hp.ov.nms.ui.framework...

   SignInOutAuditLog logSignIn:

   INFO: Successful Sign In
   User Account: system
   NNMi Role: Administrator (ADMIN)
   Remote Host: <node IP address>
   Remote Port: 1549
   Locale: en_US
   Sign In/Out Audit Since 6/14/07 9:33 AM
   =======================================
   Currently Signed In:
   #1: system <node IP address> 6/14/07 10:53 AM (last access 6/14/07
   10:53 AM)
   No users currently signed out.
   ```

To enable the audit log files:

> **Tip:** When making file changes under HA, you must make the changes on both nodes in the cluster. For NNMi using HA configurations, if the change requires you to stop and restart the NNMi management server, you must put the nodes in maintenance mode before running the ovstop and ovstart commands.

1. In a text editor, open the `nnm-logging.properties` file:
   - **Windows**:
     `%NnmDataDir%\shared\nnm\conf\props\nnm-logging.properties`
   - **UNIX**:
     `/var/opt/OV/shared/nnm/conf/props/nnm-logging.properties`
2. Search for the text block containing the following line:

   ```
   com.hp.ov.nnm.log.signin.level = OFF.
   ```
3. Modify the line to read as follows:

   ```
   com.hp.ov.nnm.log.signin.level = INFO
   ```
4. *Optional:* Add the following two configuration settings if they do not already exist:
   - Set the total number of audit log files, for example 4:
     `com.hp.ov.nnm.log.signin.count = <count value>`
   - Set the maximum size for the audit log files, for example 20M (20 megabytes):
     `com.hp.ov.nnm.log.signin.size = <file size value>`
5. Save and close the `nnm-logging.properties` file.
6. Stop `ovjboss`:

   ```
   ovstop ovjboss
   ```

   Note: The `ovstop ovjboss` command also causes the `nnmaction` process to stop.
7. Use the `ovstart` command to restart `ovjboss` and `nnmactions`:

   ```
   ovstart
   ```

Restore the Administrator NNMi Role

If you have accidentally configured NNMi so that zero NNMi users are mapped to the **NNMi User Group**[1]: NNMi Administrators (preventing anyone from being able to access the Configuration workspaces), access the NNMi console as the `system` user to correct the problem.

Sign into the console using the password that was configured for the `system` user when NNMi was first installed.

[1] NNMi User Groups are those User Groups provided by NNMi. Users cannot access the NNMi console until their User Account is mapped to at least one of the following NNMi User Groups: NNMi Administrators, NNMi Level 2 Operators, NNMi Level 1 Operators (with less access privileges than Level 2 Operators), and NNMi Guest Users

If you do not remember the password assigned to the `system` user, use the `nnmchangesyspw.ovpl` command to reset the `system` user's password.

> **Note:** If you are still unable to sign into the console, verify that the `nms-roles.properties` file is in good working order. See "Restore NNMi Access for the system User" below for more information.

Restore NNMi Access for the system User

NNMi provides an `nms-roles.properties` file that stores part of the `system` user configuration. This file is located in the following directory:

- **Windows**:
  `%NnmDataDir%\nmsas\NNM\conf\props\nms-roles.properties`
- **UNIX**:
  `/nmsas/NNM/conf/props/nms-roles.properties`

You should not need to ever modify this file.

To verify the contents of this file:

1. With a text editor, open the `nms-roles.properties` file.
2. Verify that the following required line is present:

   ```
   system = system,admin
   ```
3. Save and close the file.

Chapter 14

Configuring Incidents

Incidents are information that NNMi considers important to bring to your attention regarding your network. See "How NNMi Gathers Incidents" on the next page for more information.

NNMi provides a set of incident configurations for the following:

- Traps generated from an SNMP agent (SNMPv1, SNMPv2c, or SNMPv3)
- Syslog Messages
- Management incidents that are generated by NNMi
- Events generated by NNM 6.x or 7.x management stations

See "Incident Configurations Provided by NNMi" on page 605 for more information about the configurations provided.

Note: If a node is deleted, only an NNMi administrator can view the incidents associated with that node.

NNMi provides one centralized location, the incident views, where the management events, SNMP traps, and NNM 6.x or 7.x forwarded events are visible to your team. You control which SNMP traps and NNM 6.x or 7.x events are considered important enough to show up as incidents. You can also configure how incidents that are generated by NNMi are displayed. You and your team can easily monitor the incidents and take appropriate action to preserve the health of your network.

You can modify the incident configurations provided by NNMi or create new incident configurations. To do so, see the following topics:

> **Tip**: See "Configure a Correlation Rule" on page 682 and "Configure a Causal Rule" on page 714 for information about creating incidents for use in Custom Correlations.

- "Configure SNMP Trap Incidents" on page 782
- "Configure Syslog Message Incidents (HP ArcSight)" on page 935
- "Configure Management Events" on page 1078
- "Configure Remote NNM 6.x/7.x Events" on page 1221
- Using the Pairwise Configuration form, you can configure pairwise correlations. See "About Pairwise Configurations" on page 660 for more information.

Caution: If you make changes to an incident configuration provided by NNMi, those changes are at risk of being overwritten in the future. See Author form for important information.

You can also use the Incident Configuration form to define relationships between multiple incidents by creating deduplication and rate configurations. See "Manage the Number of Incoming Incidents" on page 653, "Correlate Duplicate Incidents (Deduplication Configuration)" on page 659, and "Track Incident Frequency (Rate: Time Period and Count)" on page 659, for more information.

You can use the Incident Configuration form to control how NNMi handles incoming SNMP traps. See "Handle Unresolved Incoming Traps" on page 776 and "Control which Incoming Traps Are Visible in Incident Views" on page 775 for more information.

Note: Each time you stop and restart ovjboss, any incidents that have not yet been correlated or persisted are lost. This means that after a restart of ovjboss, an incoming incident might not be correlated as expected. For example, after a restart of ovjboss, a duplicate incident might not be correlated under its original parent incident. Instead, a new parent incident might be generated. See "Stop or Start an NNMi Process" on page 82 for more information about starting and stopping the ovjboss process.

Manage Incidents Using Incident Configurations

NNMi enables you to control the incidents that are generated and how they are displayed. To help you manage your incidents and incident configurations, you want to understand the following:

- "How NNMi Gathers Incidents" below
- "How NNMi Closes Incidents" on page 605
- "Incident Configurations Provided by NNMi" on page 605

When managing your incidents using Incident Configurations, you can perform the following tasks:

- "Manage the Number of Incoming Incidents" on page 653
- "Track Incident Frequency (Rate: Time Period and Count)" on page 659
- "Configure an Action for an Incident" on page 748
- "Configure Diagnostics for an Incident (NNM iSPI NET)" on page 757

How NNMi Gathers Incidents

Incidents are information that NNMi considers important to bring to your attention regarding your network.

The NNMi Causal Engine analyzes the health of your network and provides the ongoing health status reading for each device. The Causal Engine also extensively evaluates problems and determines the root cause for you, whenever possible, sending incidents to notify you of problems. Any incident generated from a Causal Engine management event has an **Origin** of **NNMi** in your incident views. See Using the Incident Form for more information about incident attributes.

NNMi gathers information from the sources described in the following table.

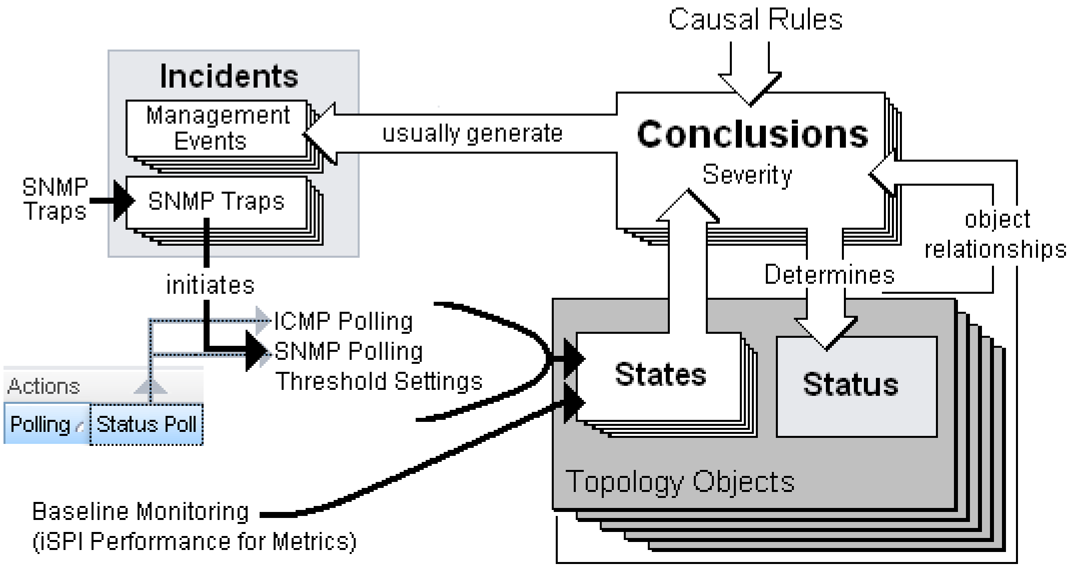
Incidents Collected by NNMi

Information Source	Description
State Poller	Tracks changes in State for an object. See Accessing Device Details for more

Incidents Collected by NNMi, continued

Information Source	Description
	information about possible States per object.
SNMP Traps	Traps are unsolicited SNMP notifications that come from your network devices. The NNMi Causal Engine uses this information as symptoms during its analysis. SNMP traps can also appear as incidents if configured to do so, using the NNMi incident configuration feature. See "Configure SNMP Trap Incidents" on page 782 for more information.
Conclusions	Every Conclusion has a Severity associated with it. The Status reported for an object is the most severe of all outstanding Conclusions. In addition, Conclusions inform the user of the underlying cause (or reason) for an object's Status. A Conclusion generates an associated Incident if it is determined to be the root cause of a problem.

Click here to view a diagram of the relationship among Conclusions, States (from State Poller), and Incidents.



See "The NNMi Causal Engine and Incidents" on the next page for an overview of what the NNMi Causal Engine does with the information collected. See "About the Event Pipeline" on page 603 for an overview of the event pipeline path each trap or NNMi event takes before NNMi creates an incident. This chronological path guarantees that the data is analyzed in chronological order.

Note: The Causal Engine also sends incident information that it generates through the event pipeline to guarantee the chronological order for determining its root cause incidents.

By default, NNMi includes preconfigured definitions for SNMP traps, Syslog Messages, NNM 6.x and 7.x events, and the incidents generated by the NNMi Causal Engine. See Incident Views Provided by NNMi for more information.

Related Topics

"Configure SNMP Trap Incidents" on page 782

"Configure Syslog Message Incidents (HP ArcSight)" on page 935

"Configure Management Events" on page 1078

"Configure Remote NNM 6.x/7.x Events" on page 1221

"Incident Configurations Provided by NNMi" on page 605

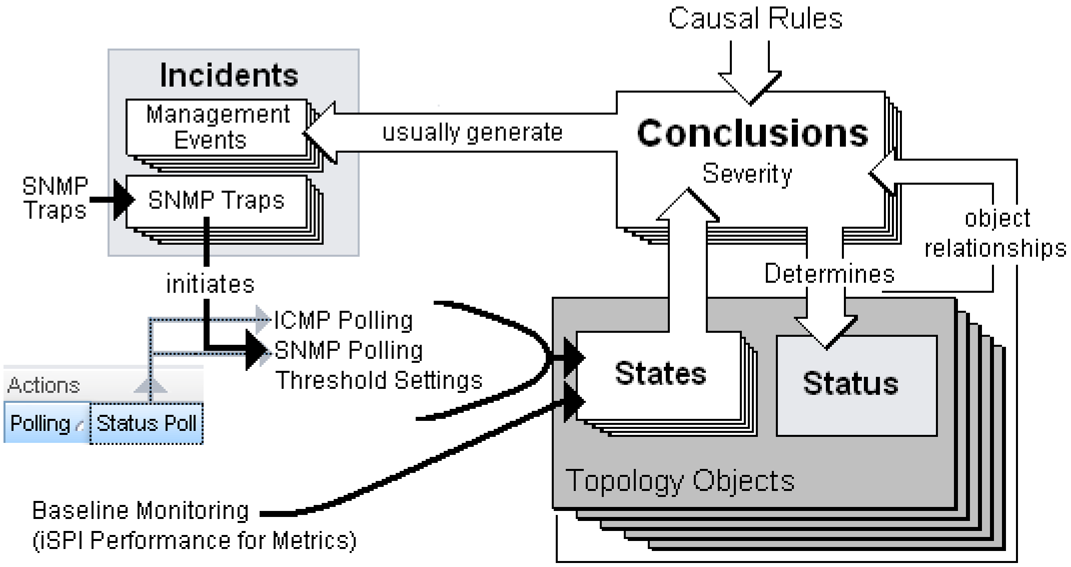
"Manage the Number of Incoming Incidents" on page 653

The NNMi Causal Engine and Incidents

The Causal Engine extensively evaluates network issues and determines the root cause for you, whenever possible, sending incidents to notify you of problems.

The NNMi Causal Engine defines root cause in terms of symptoms. To do so, it uses a set of rules to define relationships for fault and performance (thresholding) symptoms and root causes. Sources of symptom information include SNMP traps and the monitoring information from the State Poller, which includes an object's State. See "The NNMi Causal Engine and Object Status" on page 594 and "How NNMi Gathers Incidents" on page 590 for more information.

Click here to view a diagram of the relationship among Incidents, Conclusions, States, and Status.



The NNMi Causal Engine performs the following tasks:

- Generates notifications about problems.
- Generates conclusions that relate to the root cause of the problem.
- Determines whether the incident should be correlated or suppressed.

  Tip: An incident that is correlated with a Root Cause Parent Incident has a Correlation Nature of **Secondary Root Cause**. These incidents can be examined using the **All Incidents** view, but do not appear as Key Incidents or Root Cause incidents. See Incident Views Provided by NNMi for more information.

- Click here to view an incident suppression scenario.

The `AddressNotResponding` incident is suppressed by the `InterfaceDown` incident, according to the following scenario:

When an IPv4 address stops responding to ICMP, an episode begins, which exists for the duration of 60 seconds.

Within that duration, if the interface associated with that IPv4 address goes down, the Causal Engine concludes that the interface down condition caused the IPv4 address to stop responding.

Therefore, the `AddressNotResponding` incident is not generated. Only the `InterfaceDown` incident is generated.

To ensure that the `InterfaceDown` incident is detected within the duration, the Causal Engine issues a named poll for that interface. The incident enables the network engineer to fix the root cause of the problem which, in this case, is the interface.

If the interface does not go down during the episode, the Causal Engine generates an `AddressNotResponding` incident. If the interface goes down after the episode, NNMi generates the `InterfaceDown` incident. In this case, the network engineer has to treat the two problems separately.

- Click here to view an incident correlation scenario.

  The `NodeDown` incident correlates the `InterfaceDown` incident from one-hop neighbor interfaces, according to the following scenario:

  When an interface goes down, a `NodeDown` episode begins for the neighboring node, which exists for the duration of 300 seconds.

  Within that duration, if the node goes down, the `InterfaceDown` incident is correlated with the `NodeDown` incident.

  The `InterfaceDown` incidents from all one-hop neighbors are correlated with the `NodeDown` incident. The network operator can review the `InterfaceDown` incidents as supporting evidence for the `NodeDown` incident.

- Closes incidents that are no longer valid (for example, when a "Cold Start" trap is received a short time after a "Node Down" incident was generated because a device was recently rebooted).
- Creates a parent-child relationship between incidents that are all related to one problem (for example, a "Node Down" incident contains a child "Interface Down" incident for each neighboring interface of the node).
- Creates parent-child relationships between incidents that are correlated using the Custom Correlation configuration. NNMi's Custom Correlation feature enables administrators to add customized rules for when and how to correlate incidents. See "Configure Custom Correlations" on page 680 for more information.

The Causal Engine actively solicits symptoms during analysis and reacts dynamically to topology changes. The Causal Engine uses the following three stages to help determine and display root cause incidents and their related conclusions.

NNMi Causal Engine Stages

Causal Engine Stages	Description
Condition Listener	Collects symptoms from NNMi processes and services.
Hypothesis engine	Analyzes these symptoms to determine relationships until a root cause is reached.
Blackboard	Based on the information sent by the hypothesis engine, the blackboard updates a device's status and posts any related incidents.

The NNMi Causal Engine analyzes the health of your network and provides the ongoing health Status reading for each object it monitors. See "The NNMi Causal Engine and Object Status" belowand "The NNMi Causal Engine and Monitoring" on page 343 for more information.

Related Topics

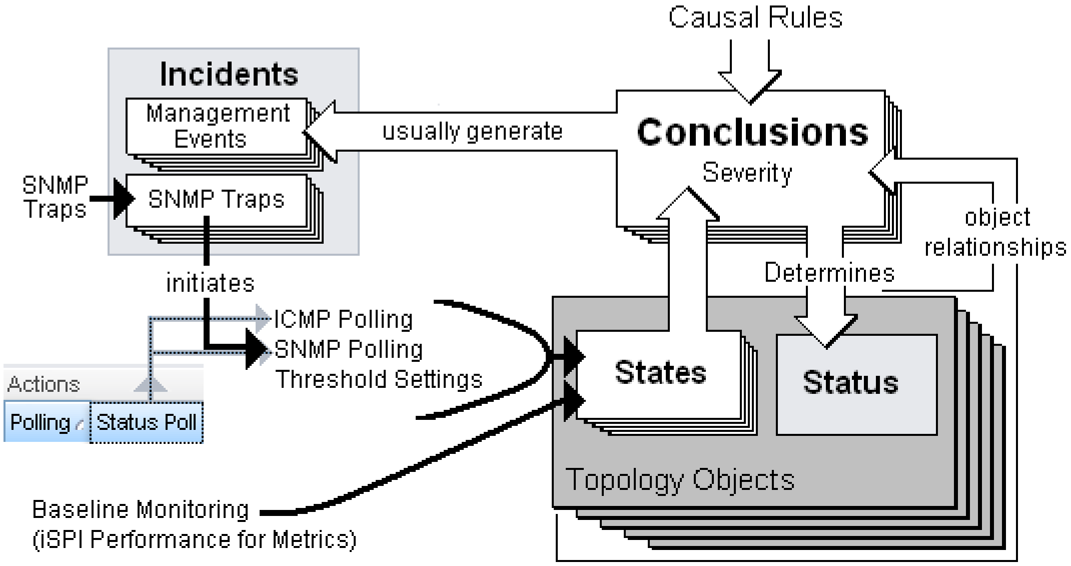
"The NNMi Causal Engine and Object Status" below

The NNMi Causal Engine and Object Status

The Causal Engine sets the Status on relevant network objects. Status indicates the overall health of an object and is determined from the outstanding Conclusions. Every Conclusion has a Severity associated with it. The Status reported is the most severe of all outstanding Conclusions. In addition, Conclusions inform the user of the underlying cause (or reason) for an object's Status.

See the Conclusion Tab information for each object form in Accessing Device Details for information about possible Conclusions for each NNMi object.

Click here to view a diagram of the relationship among Incidents, Conclusions, States, and Status.



The Causal Engine uses the following Status categories in decreasing order of severity:

? Unknown

▨ Disabled

⊗ Critical

▽ Major

⚠ Minor

▲ Warning

✓ Normal

⊘ No Status

To determine why an object is not polled (No Status), do the following:

- Select the object from the table or map view and access **Actions** >:**Configuration Details** > **Monitoring Settings** .
- Select the node of interest or the node that is hosting the object of interest from the table or map view and access **Actions** >:**Configuration Details** > **Communication Settings**.

NNMi analyzes a variety of network objects using either the SNMP protocol or ping to retrieve information about the network object. The following list shows the network objects that NNMi monitors and analyzes. Click each object for more information.

- **Aggregator Interface**

  An Aggregator Interface is a set of interfaces on a switch that are linked together, usually for the purpose of creating a trunk (high bandwidth) connection to another device. Aggregator Interfaces have designated Aggregator Member Interfaces.

  NNMi reports that Status of an Aggregator Interface as follows:

  ? Unknown - The Status of all Aggregation Members of the Aggregator Interface are Unknown.

  ⊗ Critical - The Aggregator Interface, or all of the Aggregation Members, or both are operationally down. This means `ifOperStatus` is `down`.

  ⚠ Minor - Some Aggregation Members (but not all Aggregation Members) of the Aggregator Interface are operationally down. This means the `ifOperStatus` is `down`.

  ✓ Normal - All Aggregation Members of the Aggregator Interface are operationally up. This means `ifOperStatus` is `up`.

  ⊘ No Status - All Aggregation Members of the Aggregator Interface are not polled.

- **Aggregator Layer 2 Connection**

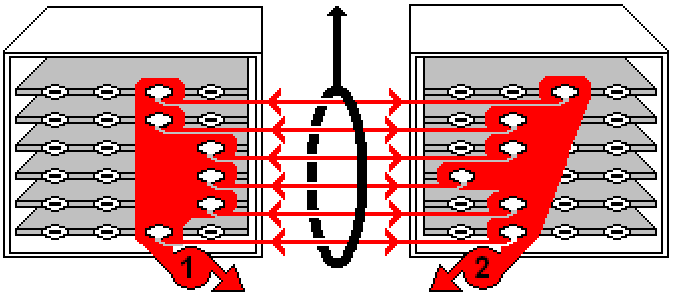
  An Aggregator Layer 2 Connection is a connection with endpoints that are Aggregator Interfaces. These are usually high bandwidth connections that link switches. Aggregator Layer 2 Connections have Aggregator Interfaces and Aggregation Members.

Click here to see a diagram of an example Link Aggregation.

Example Link Aggregation

Thick Line on Layer 2 Neighbor View Map =
one Aggregator Layer 2 Connection:

- Logical unit (not physical)
- Functions as if it were one
- 6 Aggregation Member Layer 2 Connections



two Aggregator Interfaces:

- Logical units (not physical)
- Each functions as if it were one
- Each has 6 Aggregation Member Interfaces

NNMi reports the Status of an Aggregator Layer 2 Connection as follows:

Unknown - The Status of any Aggregation Member of the Aggregator Layer 2 Connection is Unknown.

Critical - The Aggregator Interface, the Aggregation Member, or both are operationally down. This means `ifOperStatus` is `down`.

Minor - Some Aggregation Members, but not all, are operationally down. This means `ifOperStatus` is `down`.

Normal - All Aggregation Members of the Aggregator Layer 2 Connection are operationally up. This means `ifOperStatus` is `up`.

No Status - All Aggregation Members of the Aggregator Layer 2 Connection are not polled.

- **Card**

  A Card is a physical component on a device that has physical ports that contain one or more interfaces used to connect to other devices. A Card can also contain sub-cards. The Card containing another Card is known in NNMi as the Parent Card. The sub-card is known as the Daughter Card. NNMi supports Daughter Cards one-level deep.

  NNMi reports the status of a Card as follows:

  Unknown - Indicates the SNMP Agent associated with the Card does not respond to SNMP queries.

  Disabled - The Card or Daughter Card is administratively down or disabled . This means the `cardAdminStatus` is `down`.

  Critical - The Card is operationally down. This means the `cardOperStatus` is `down`.

Minor - The Card is neither up nor down. This means the `cardOperStatus` is `unknown` or `other`.

Normal - The Card is operationally up. This means the `cardOperStatus` is **`up`**.

No Status - The Card is not polled.

- **Card Redundancy Group**

  A Card Redundancy Group is a set of card modules that are configured to provide Card redundancy on the device. NNMi supports two Cards in a Card Redundancy Group. One Card in the Card Redundancy Group acts as the Primary member. The other Card acts as the Secondary member. If the Primary card fails, the Secondary Card takes over as the Primary Card.

  NNMi reports the Status of Card Redundancy Groups as follows:

  Unknown - All cards in the Card Redundancy Group have an Unknown Status.

  Critical - Indicates either of the following:

  - No Card is acting as the Primary member of the Card Redundancy Group.
  - Both Cards are acting as the Primary member of the Card Redundancy Group.

  Major - At least one card in the group is reporting a state that does indicates it is neither the Primary or Secondary card.

  Warning - The Card Redundancy Group has no Secondary member.

  Normal - The Card Redundancy Group is functioning correctly.

  No Status - The Card Redundancy Group has not yet been discovered or is not being polled.

- **Connections**

  Note: Connections on Layer 3 maps never have status.

  Connections are Layer 2 physical connections and Layer 3 network connections. NNMi discovers connection information by reading forwarding database (FDB) tables from network devices and gathering data from a variety of Layer 2 *discovery protocols* (see the list of Topology Source protocols in Layer 2 Connection Form).

  NNMi reports the Status of Connections as follows:

  Unknown – All endpoints of the connection have unknown status.

  Disabled – Any one endpoint of the connection is disabled.

  Critical– All endpoints are operationally down.

  Minor – Any one endpoint is down.

  Warning – Endpoints have unknown and non-critical Status.

  Normal – All endpoints are operationally up.

  No Status – All endpoints are not polled.

Note: Pseudo interfaces do not affect Connection Status. See Interfaces (All Attributes) View (Inventory) for more information about pseudo interfaces.

- **Field Replaceable Units (FRU Card)**

  A Field Replaceable Unit (FRU) Card is a Card that can be replaced on a device that is operationally active (not powered down). When an FRU card is removed from or added to the device, NNMi reports the occurrence with an incident. If an FRU Card is not recognized by the device, NNMi reports the unrecognized Card with an incident.

  NNMi reports the Status of an FRU Card as follows:

  Unknown - Indicates either of the following:

  - The SNMP Agent associated with the card does not respond to SNMP queries.
  - NNMi cannot determine the `cardOperStatus` or `cardAdminStatus` values.

  Disabled - The Card is administratively down. This means the `cardAdminStatus` is `down`.

  Critical - The Card is operationally down. This means the `cardOperStatus` is `down`.

  Minor - The Card is neither up nor down. This means the `cardOperStatus` is either `unknown` or `other`.

  Normal - The Card is operationally up. This means the `cardOperStatus` is `up`.

  No Status - The Card is not being polled.

- **Interface**

  An interface is a logical object that might or might not be associated with a physical port. Interfaces are used to identify connections between nodes. Multiple interfaces can be associated with a single physical port. NNMi identifies interfaces using either of the following values:

  - ifName
  - ifAlias
  - ifType[ifIndex] (for example, ethernetCsmacd[17])

  Each physical port managed by NNMi is associated with one or more interfaces. NNMi identifies ports using the <*Card-number* / *Port-number*> value.

  NNMi reports the Status of Interfaces as follows:

  Unknown - Indicates either of the following:

  - The SNMP Agent associated with the interface does not respond to SNMP queries.
  - NNMi cannot determine the health because `ifAdminStatus` and `ifOperStatus` cannot be measured.

  Disabled - Interface is administratively down. This means `ifAdminStatus` is `down`.

  Critical - Interface is operationally down. This means `ifOperStatus` is `down`.

  Normal - Interface is operationally up. This means `ifOperStatus` is `up`.

  No Status - Interface is not polled.

- **IP Address**

  An IP address is a routable address that responds to ICMP. IP addresses are typically associated with nodes.

  NNMi reports the status of a IP Addresses as follows:

  Disabled - The interface associated with this IPv4 address is administratively down or disabled.

  Critical - IP address does not respond to ICMP queries (ping the device).

  Normal - IP address responds to ICMP queries.

  No Status - IP address is not polled.

- **Node**

  A node is a device that NNMi finds as a result of the Spiral Discovery process. A node can contain interfaces, boards, and ports. You can separate nodes into two categories:

  - Network nodes, which are active devices such as switches, routers, and hubs
  - End nodes, such as UNIX or Windows servers

  NNMi typically manages network nodes, reporting Status as follows:

  Unknown – Indicates SNMP node is unresponsive due to either of the following circumstances:

  - The SNMP Agent associated with the node does not respond to SNMP queries and the polled IP addresses do not respond to ICMP queries
  - The polled IP addresses associated with the non-SNMP node does not respond to ICMP queries

  Disabled - Indicates a neighbor interface has been disabled, causing the node to be unreachable.

  Critical – Indicates any one of the following:

  - The node is down as determined by neighbor analysis.
  - The node is marked as important and is unresponsive (NNMi cannot access the node from the NNMi management server).
  - The node is unconnected (it has no neighbors) and, therefore, is unresponsive.
  - NNMi cannot determine if the node is down or if the incoming connection is down.
  - At least one Custom Polled Instance associated with the node has a Status of Critical and Custom Polled Instances are configured to affect Node Status.

  Minor - A managed object in the Node has any of the following problems:

  - The SNMP Agent associated with the Node does not respond to SNMP queries.
  - The management address on the Node is not responding to ICMP.

  - One or more interfaces on the Node are operationally down. This means `ifOperStatus` is `down`.
  - One or more IP addresses on the Node do not respond to ICMP.
  - NNMi is unable to measure the Status of one or more Cards on the Node. This means the `cardOperStatus` is either `unknown` or `other`.
  - At least one Interface on the Node has a threshold outside the range specified for the device.
  - At least one Custom Polled Instance associated with the Node has a Status of Minor and Custom Polled Instances are configured to affect Node Status.
  - One or more cards in the Node are operationally down. This means `cardOperStatus` is `down`.

  ▲ Warning - A managed object on the Node has any of the following problems:

  - At least one Card in a Card Redundancy Group associated with the Node is malfunctioning.
  - At least one Custom Polled Instance associated with the Node has a Status of Warning and Custom Polled Instances are configured to affect Node Status.

  ▽ Major - Indicates NNMi detected any of the following:

  - A fan (Node Component) failure
  - A power supply (Node Component) failure
  - A backplane (Node Component) failure
  - A memory (Node Component) failure
  - At least one Custom Polled Instance associated with the Node has a Status of Major and Custom Polled Instances are configured to affect Node Status.

  ✓ Normal - All objects associated with the node are operationally up.

  ⊘ No Status – The SNMP Agent, all interfaces, and all IP addresses of the node are not polled.

- **Node Components**

  Large (or more sophisticated) network devices often require special environments and components to function properly. Examples are power supplies, fans, voltage regulators, and internal computers. These device components can be monitored by component health sensors. An administrator can monitor the health of these components to know when any of them has failed or is operating marginally.

  NNMi reports the status of Node Components as follows:

  ⊗ Critical – The component is not functioning properly.

  ✓ Normal – The component is operating properly

  ⊘ No Status – The component is not polled.

- **Node Groups**

  A Node Group is a logical collection of nodes for customize polling configuration. An administrator creates Node Groups. .

An NNMi administrator can also configure Node Group Status calculations. The out-of-the-box configuration propagates the most severe Status as follows:

Critical – At least one node in the Node Group has Critical Status.

Major – No nodes have a Critical Status, and at least one node in the Node Group has Major Status.

Minor – No nodes in the Node Group have Critical or Major Status, and at least one Node in the Node Group has Minor Status.

Warning – No nodes in the Node Group have Critical, Major, or Minor Status, and at least one Node in the Node Group has Warning Status.

Normal – No nodes in the Node Group have Critical, Major, Minor, or Warning status, and at least one Node in the Node Group has Normal Status.

Unknown – No nodes in the Node Group have Critical, Major, Minor, Warning, or Normal Status, and at least one Node in the Node Group has Unknown Status.

No Status – All nodes in the group have No Status.

- **Redundant Router Groups**

  A Router Redundancy Group is a set of routers that are configured to provide redundancy in the network. For example, groups of routers configured using protocols such as:

  - Hot standby router protocol (HSRP)
  - Virtual router redundancy protocol (VRRP)

  Router Redundancy Groups usually have a single device acting as the Primary device, a single device acting as a Secondary device, and any number of Standby devices. If the Primary device fails, the Secondary device takes over as Primary, and one of the Standby devices becomes Secondary.

  NNMi reports the Status of Router Redundancy Groups as follows:

  Critical – The Router Redundancy Group has no acting Primary router.

  Major – The Router Redundancy Group's Primary device is not properly configured (for example, multiple Primary routers exist).

  Minor – The Router Redundancy Group' Secondary device is not properly configured (for example, no acting Secondary router exists).

  Warning – The Router Redundancy Group is functioning, but is in some way degraded.

  Normal – The Router Redundancy Group is functioning properly.

  No Status – The Router Redundancy Group is not yet fully discovered or populated.

- **SNMP Agent**

  An SNMP agent is a process running on the managed node, which provides management functions. The SNMP agent is responsible for managing interfaces and ports on the managed node. It can be associated with one or more nodes.

  NNMi reports the Status of SNMP Agents as follows:

✖ Critical - SNMP Agent does not respond to SNMP queries.

⚠ Minor - The address associated with the SNMP Agent is not responding to ping.

△ Warning - A high or abnormal ICMP response time from the NNMi management server to the selected node is reported.

✓ Normal - SNMP Agent responds to SNMP queries.

⊘ No Status - SNMP Agent is not polled.

Related Topics

"The NNMi Causal Engine and Incidents" on page 592

About the Trap Service Stages

Any trap information that appears in the NNMi console or in an NNMi log file is first processed through the NNMi Trap Service. The NNMI Trap Service guarantees that the trap data is analyzed in chronological order.

The following table describes the NNMi Trap Service stages.

NNMi Trap Service Stages

Trap Service Stages	Description
SnmpTrapListener	Receives traps from the configured "Listen" interface. No filtering takes place at this stage.
MessageProcessor	Parses raw traps and records traps for audit purposes. If Trap Logging is enabled, the MessageProcesser writes all traps to the trap log. **Note**: Traps configured in `trapFilter.conf` file are not written to the log file.
TrapServerConfiguration	Handles configuration updates.
NarrowTrapAnalysis	Handles Hosted Object Trap Storm detection and suppression. **Note**: This stage is disabled by default. To enable this state use: `nnmtrapconfig.ovpl -setProp hostedObjectTrapstorm true -persist` See Hosted Object Trap Storm for more information about Hosted Object Trap Storm incidents.
WideTrapAnalysis	Handles Trap Storm detection and suppression. **Note**: This stage is enabled by default. See Trap Storm for more information about Trap Storm incidents.
TrapFilter	Drops all traps that are older than 10 minutes or blocked by IPAddress and OID. **Note**: Use `nnmtrapd.conf` to configure trap filters.

NNMi Trap Service Stages, continued

Trap Service Stages	Description
	This filter only passes traps that are configured and enabled in the SNMP Trap Incident Configuration workspace.
TrapServerConfiguration	Forwards traps to the Events Pipeline. This stage also handles Hosted Trap Storm and Trap Storm incident generation. See Hosted Object Trap Storm and Trap Storm for more information.
ForwardingStage	Forwards traps to another destination, if specified. For example, traps might be forwarded to another instance of NNMi or to other integrated software.

About the Event Pipeline

Any incident information that appears in your incident views first travels through the event pipeline. The event pipeline guarantees that the incident data is analyzed in chronological order.

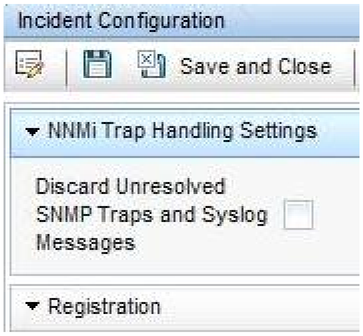
Note: Not all information that travels through the pipeline results in an incident.

If an incident does not meet the criteria for an event pipeline stage, it is ignored and passed to the next stage in the pipeline. The following table describes the event pipeline stages.

NNMi Event Pipeline Stages

Event PipelineStages	Description
SNMP Trap and Event Receiver	Accepts all SNMP traps. **Tip**: See "About the Trap Service Stages" on the previous pagefor information about Trap Service stages that occur before the Event Pipeline stages begin.
pmd Receiver	Accepts NNM events forwarded from remote NNM 6.X and 7.X management stations.
Incident Receiver	Accepts all incident information that comes from the NNMi Causal Engine. See "The NNMi Causal Engine and Incidents" on page 592 **Note**: The incident information that is received includes any Custom Correlation configurations.
Geo Incident Receiver	Accepts all incident information that comes from Global or Regional Managers.
Type Enforcer	Determines if a configuration exists for this trap, event, or incident. If the incident configuration exists, the type enforcer begins to populate the incident fields according to the configuration. Examples of the incident fields that are populated include **Severity**, **Origin**, **Category**, and **Correlation Nature**. If an incident configuration is disabled or does not exist for the incident, NNMi drops the incident.

NNMi Event Pipeline Stages, continued

Event PipelineStages	Description
Resolver	Drops the trap if the Source Object or Source Node is not in the topology, unless the "Discard Unresolved SNMP Traps and Syslog Messages" check box is unchecked.  Incident Configuration Save and Close NNMi Trap Handling Settings Registration Discard Unresolved SNMP Traps and Syslog Messages □ Determines if the incident's Source Node or Source Object (such as interface or card) matches an object in the NNMi database. If available, the Resolver populates the incident with the most current Source Node and Source Object attribute values.
Customization	Checks for any of the following incident configurations in the order listed: • Suppression • Enrichment • Dampening
Store Bulk	Collects incidents and stores them. NNMi stores this information in bulk, using a pre-defined time period or number of incidents, whichever occurs first. The default time period is 3 seconds. The default number of incidents is 300.If you send a trap and subsequent traps do not occur on the network for a period of time after the trap is sent, NNMi waits up to 30 seconds before persisting new incident or trap information.
Notification	Notifies other process and services about a new incident.
Pairwise	Checks for any current pairwise configurations for the incident.
Rate	Checks for any current rate configurations for the incident.
Dedup	Checks for any current deduplication configurations for the incident.
Relate	Performs any additional Causal Engine correlations, including Custom Correlations, and cancels the incident when applicable.
Actions	Performs any automatic actions that the NNMi administrator has configured to be run for one or more incidents. See Using Actions to Perform Tasks for more information.

NNMi Event Pipeline Stages, continued

Event PipelineStages	Description
Rba	(NNM iSPI NET only) The Diagnostics stage. Checks whether Diagnostics should be run on the current incident and submits a execution request to run the Diagnostics report on the device.

How NNMi Closes Incidents

NNMi closes incidents under the following circumstances:

- The incident's configuration is a Pairwise Configuration and both incidents specified in the pair occurred in the order specified. See "About Pairwise Configurations" on page 660 for more information.
- NNMi determines that the problem that generated the incident is resolved. For example, NNMi closes a Down incident when a Conclusion indicates the node or device is available for use and has returned to a normal state for a specified threshold of time.

  See "Incident Configurations Provided by NNMi" below for the incident configurations that NNMi provides.

An NNMi administrator can also manually change the incident Lifecycle State to Closed. An operator might also be able to change the incident Lifecycle State to Closed if the NNMi administrator chooses to make this Action available to operators.

Note the following:

- If a node is deleted, NNMi closes the incident.
- The NNMi Causal Engine does not generate Conclusions during initial discovery.
- NNMi only Closes incidents for those objects that have one or more outstanding Conclusions as indicated in the object form's Conclusions tab.

Incident Configurations Provided by NNMi

NNMi provides several incident configurations out-of-the-box. You can review these configurations or modify these configurations to better meet your needs. For example, you might want to customize the message that appears with a particular type of incident, including adding information to the message displayed.

You might also choose to create your own configurations for additional SNMP traps that are important to you.

These out-of-the-box configurations are organized according to the following categories:

"SNMP Trap Incident Configurations Provided by NNMi" on page 612

"Syslog Message Incident Configurations Provided by NNMi" on page 622

"Management Event Configurations Provided by NNMi" on page 632

"Remote NNM 6.x/7.x Event Configurations Provided by NNMi" on page 629

"Incident Pair (Pairwise) Configurations Provided by NNMi" on page 661

Caution: If you make changes to an incident configuration provided by NNMi, those changes are at risk of being overwritten in the future. See Author form for important information.

Custom Incident Attributes Provided by NNMi (Information for Administrators)

NNMi uses custom incident attributes to attach additional information to incidents.

A subset of CIAs is available for any particular incident. Any relevant CIAs are displayed in the Incident form, on the Custom Attributes tab. There are two categories of possible CIAs:

1. **Custom incident attributes**
   - Provided by NNMi
   - Provided for NNM iSPI Performance for Metrics
2. **SNMP trap varbinds**
   - Identified by the Abstract Syntax Notation value (ASN.1). Varbinds are defined in MIB files that you can load into NNMi. See "Load SNMP Trap Incident Configurations" on page 771.

The following tables explain the custom incident attributes provided by NNMi.

Custom Incident Attributes Provided by NNMi

Name	Description

cia.address	This attribute value is determined by the `com.hp.nnm.trapd.useUdpHeaderIpAddress`property defined in the following file:
	Windows:
	`%NNM_DATA%\shared\nnm\conf\props\nms-jboss.properties`
	UNIX:
	`$NNM_DATA/shared/nnm/conf/props/nms-jboss.properties`
	When `com.hp.nnm.trapd.useUdpHeaderIpAddress=true`, the cia.address value is the User Datagram Protocol (UDP) header IP Address.
	When `com.hp.nnm.trapd.useUdpHeaderIpAddress=false`, both the cia.address and cia.originaladdress values contain the SNMP Agent IP Address. The `com.hp.nnm.trapd.useUdpHeaderIpAddress` property is false by default.
	See the "Maintaining NNMi" chapter in the *HP Network Node Manager i Software Deployment Reference* for more information.

Custom Incident Attributes Provided by NNMi, continued

Name	Description

cia.originaladdress	This attribute value is determined by the `com.hp.nnm.trapd.useUdpHeaderIpAddress`property defined in the following file:
	Windows:
	`%NNM_DATA%\shared\nnm\conf\props\nms-jboss.properties`
	UNIX:
	`$NNM_DATA/shared/nnm/conf/props/nms-jboss.properties`
	This Custom Incident Attribute enables you to access both the User Datagram Protocol (UDP) header IP Address and the SNMP Agent IP Address of the managed device.
	When `com.hp.nnm.trapd.useUdpHeaderIpAddress=true`, cia.originaladdress is the value of the SNMP Agent IP Address and the cia.address value is the User Datagram Protocol (UDP) header IP Address.
	When `com.hp.nnm.trapd.useUdpHeaderIpAddress=false`, both cia.originaladdress and cia.address values contain the SNMP Agent IP Address. The `com.hp.nnm.trapd.useUdpHeaderIpAddress` property is false by default.
	See the "Maintaining NNMi" chapter in the *HP Network Node Manager i Software Deployment Reference* for more information.

cia.agentAddress	The IP Address that is stored in the SNMPv1 trap data for the SNMP Agent that generated the trap.
cia.custompoller.mibInstance	Instance number used to identify the row in the MIB table that contains the MIB value. **Tip**: You can use this CIA in the Message Format for a Custom Poller incident.
cia.custompoller.instanceDisplayValue	Value that results from the Instance Display Configuration. **Tip**: You can use this CIA in the Message Format for a Custom Poller incident. See "MIB Expressions Form (Custom Poller)" on page 431 for more information.
cia.custompoller.instanceFilte	The instance of the MIB Variable after the MIB Filter is applied

Custom Incident Attributes Provided by NNMi, continued

Name	Description
rValue	to the nodes in the specified Node Group. **Tip**: You can use this CIA in the Message Format for a Custom Poller incident. The MIB Filter Variable is specified when configuring a Custom Poller Collection. The MIB Filter is specified when configuring a Custom Poller Policy for the collection. See "Create a Custom Poller Collection" on page 421and "Create a Policy" on page 449 for more information.
cia.cardsRemoved	Comma-separated list of removed card names used for formatting the **Card Removed** incident message.
cia.cardsInserted	Comma-separated list of the inserted card names used for formatting the **Card Inserted** incident message.
cia.cardsRemoved	Comma-separated list of removed card names used for formatting the **Card Removed** incident message.
cia.cardsInserted	Comma-separated list of the inserted card names used for formatting the **Card Inserted** incident message.
cia.custompoller.collection	The Name of the associated Custom Poller Collection.
cia.custompoller.lastValue	The last polled value that caused a state change which generated the incident.
cia.custompoller.policy	The Name of the associated Custom Poller Policy.
cia.custompoller.variable.description	The description of the MIB expression being polled.
cia.custompoller.variable.expression	The MIB expression that was collected and the computed value that caused the incident.
cia.custompoller.variable.name	The Name of the MIB expression variable that caused the incident.
cia.custompoller.state	The state of the Custom Polled Instance for this incident.
cia.incidentDurationMs	The time measured in milliseconds between when NNMi detected a problem with one or more network devices to the time the problem was resolved. Use this CIA to track the total time a particular object in the network was down or unavailable. **Note**: This CIA is used only when NNMi's Causal Engine has analyzed and Closed the incident. Any time an incident is closed manually (for example, by the network operator), NNMi does not include cia.incidentDuration.

Custom Incident Attributes Provided by NNMi, continued

Name	Description
cia.internalAddress	If *static* Network Address Translation (NAT) is part of your network management domain, and the NNMi management server is outside of that static NAT domain, the NNMi administrator can configure this attribute to show the internal IP address that is mapped to the external management address of the selected incident's Source Node. **Note**: The external management IP addresss (public address) must be mapped to this internal address (such as private IPv4 address) using the Overlapping IP Address Mapping Form. See "Overlapping Address Mapping Form" on page 192 for more information. For more information about Overlapping IP Addresses in an NNMi network see "Overlapping Address Mapping" on page 191.

cia.island.name	Name NNMi uses to identify the nodes contained in the island.
	NNMi administrators can use this cia value in Launch Actions to display the associated table view or topology map.
	To launch the associated topology map, use the following syntax for the Launch Action **Full URL** attribute value:
	`/nnm/launch?cmd=showNodeGroup&name=${cias[name=cia.island.name].value`
	To launch the associated table view, use the following syntax for the Launch Action **Full URL** attribute value:
	`/nnm/launch?cmd=showView&view=allNodesTableView&nodegroup=${cias[name=cia.island.name].value}`
	See "Configure Launch Actions" on page 1422 and "Attributes per Object Type for Full URLs" on page 1426 for more information.

cia.island.numberOfNodes	Number of nodes contained in the island. Use this number to determine the effect of the associated Island Down incident. See Island Group Down for more information.
cia.reasonClosed	The Conclusion information identifying the reason NNMi changed the incident's Lifecycle State to Closed. For example, NNMi might include an Interface Up Conclusion as the reason an Interface Down incident was closed. **Note**: This CIA is used when NNMi's Causal Engine has analyzed and Closed the incident. Software that is integrated with NNMi might also provide values for cia.reasonClosed. Any time an incident is closed manually (for example, by the network operator), NNMi

Custom Incident Attributes Provided by NNMi, continued

Name	Description
	does not include cia.reasonClosed.
cia.remotemgr	Hostname or IP address of the (*NNMi Advanced - Global Network Management feature*) NNMi Regional Manager that is forwarding the event
cia.securityGroup.name	Name value for the Security Group. See "Configure Security Groups (Security Group Form)" on page 560 for more information. **Note**: This CIA does not appear if the node is assigned to the Default Security Group provided by NNMi.
cia.securityGroup.uuid	UUID value for the Security Group. See "Configure Security Groups (Security Group Form)" on page 560 for more information. **Note**: This CIA does not appear if the node is assigned to the Default Security Group provided by NNMi.
cia.snmpoid	SNMP trap object identifier.
cia.sourceNodeLongName	Fully qualified DNS name for the incident's Source Node.
cia.tenant.name	Name value for the Tenant. See "Use the Tenant Form" on page 196 for more information. **Note**: This CIA does not appear if the node is assigned to the Default Tenant provided by NNMi.
cia.tenant.uuid	UUID value for the Tenant. See "Use the Tenant Form" on page 196 for more information. **Note**: This CIA does not appear if the node is assigned to the Default Tenant provided by NNMi.
cia.timeIncidentDetectedMs	The timestamp in milliseconds when NNMi first detected the problem associated with an incident. **Note**: This CIA is used only when NNMi's Causal Engine has analyzed and Closed the incident. Any time an incident is closed manually (for example, by the network operator), NNMi does not include cia.timeIncidentDetected.
cia.timeIncidentResolvedMs	The time when NNMi determines the problem associated with the incident is resolved. **Note**: This CIA is used only when NNMi's Causal Engine has analyzed and Closed the incident. Any time an incident is closed manually (for example, by the network operator), NNMi does not include cia.timeIncidentResolved.

For network performance monitoring, additional custom incident attributes are provided for your use. Click here for more information.

Many incidents are candidates for these custom incident attributes:

Information about configuring thresholds is in the following topics:

- "Configure Threshold Monitoring for Node Groups" on page 402
- "Configure Threshold Monitoring for Interface Groups" on page 381
- "Configure Threshold Information for a Custom Poller Collection" on page 441

Custom Incident Attributes Provided for Thresholding

Name	Description
cia.thresholdParameter	The Monitored Attribute that is being measured in the threshold's configuration settings. For example, **Input Utilization**.
cia.thresholdLowerBound	The configured value that when *crossed* indicates a low threshold situation.
cia.thresholdUpperBound	The configured value that when *crossed* indicates a high threshold situation.
cia.thresholdPreviousValue	Threshold results from the previous Polling Interval. For example, the threshold results might change from **Nominal** to **High**, based on a change in the `cia.thresholdMeasuredValue`. See Interface Form: Performance tab for a list of additional example Threshold result values.
cia.thresholdCurrentValue	Threshold results from the most recent Polling Interval. For example, **High**.
cia.thresholdMeasuredValue	The most recent value of the Measured Attribute being monitored according to this threshold's criteria settings. This measurement is the average of all measurements taken during the last polling interval (determined by the NNMi State Poller).
cia.thresholdMeasurementTime	The time at which the threshold was *crossed*. The time appears in ISO 8601 format.

These CIAs are used in a variety of ways:

- In SNMP trap configurations. See "Configure SNMP Trap Incidents" on page 782.
- In management events. See "Configure Management Events" on page 1078.
- In automatic actions. See "Configure an Action for an Incident" on page 748.
- In correlation configurations. See "Manage the Number of Incoming Incidents" on page 653.
- In Launch Action definitions (access through the Actions menu). See "Control the NNMi Console Menus" on page 1414.

SNMP Trap Incident Configurations Provided by NNMi

Caution: If an SNMP Trap Incident configuration's **Author** value is **HP Network Node Manager**, it can be overwritten by NNMi. See Author form for important information.

NNMi provides the SNMP trap incident configurations described in the following table.

You might also choose to create your own configurations for additional SNMP traps that are important to you.

SNMP Trap Configurations Provided by NNMi

Incident Configuration Name	Description
BGPBackward Transition	Generated when the BGP Finite State Machine moves from a higher numbered state to a lower numbered state.
BGPEstablished	Generated when the BGP Finite State Machine enters the ESTABLISHED state.
CempMemBufferNotify	Signifies that a cempMemBufferPeak object has been updated in the buffer pool.
CiscoChassisAlarmOff	Signifies that the agent entity has detected the chassisTempAlarm, chassisMinorAlarm, or chassisMajorAlarm object in this MIB has transitioned to the off(1) state.
CiscoChassisAlarmOn	Signifies that the agent entity has detected the chassisTempAlarm, chassisMinorAlarm, or chassisMajorAlarm object in this MIB has transitioned to the on(2) state.
CiscoChassisChangeNotification	Agent detects any hot-swap component change or changes in the chassis.
CiscoColdStart	Occurs when a Cisco Agent is powered up.
CiscoDemand NeighborLayer2Change	Sent to the manager whenever the D-channel of an interface changes state.
CiscoEnvMonFanNotification	Indicates at least one of the fans in the fan array has failed.
CiscoEnvMonFanStatusChange Notification	Indicates a state change for a device being monitored by ciscoEnvMonFanState.
CiscoEnvMonRedundantSupplyNotifcation	Indicates the redundant power supply failed.
CiscoEnvMonSuppStatusChangeNotification	Indicates a change in the state of a device being monitored by ciscoEnvMonSupplyState.
CiscoEnvMonTemperatureNotification	Indicates the temperature measured at a given

SNMP Trap Configurations Provided by NNMi, continued

Incident Configuration Name	Description
	testpoint is outside the normal range for the testpoint, For example, it is at the warning, critical, or shutdown stage.
CiscoEnvMonTempStatusChangeNotification	Indicates a change in the state of a device being monitored by ciscoEnvMonTemperatureState.
CiscoEnvMonVoltageNotification	Indicates the voltage measured at a given testpoint is outside the normal range for the testpoint. For example, it is at the warning, critical, or shutdown stage.
CiscoEnvMonVoltStatusChangeNotification	Indicates a change in the state of a device being monitored by ciscoEnvMonVoltageState.
CiscoFRUInserted	Indicates a Field Replaceable Unit (FRU) was inserted into the source node.
CiscoFRURemoved	Indicates a Field Replaceable Unit (FRU) was removed from the source node.
CiscoLinkDown	Occurs when the Cisco agent detects an interface has gone down.
CiscoLinkUp	Occurs when the Cisco agent detects an interface has come back up.
CiscoModuleDown	Signifies that the SNMP Agent has detected that the card has gone down.
CiscoModuleStatusChange	Indicates the Operational State of the card has changed.
CiscoModuleUp	Signifies that the SNMP Agent has detected that the card has come back up.
CiscoRFProgressionNotif	Notification sent by the active Card (for example Card Active), whenever its Redundancy Framework (RF) state changes or the RF state of the second card in the Card Redundancy Group changes.
CiscoRFSwatcNotif	Sent by the newly Active Card (for example Card Active). Indicates that a card state has been switched to a different state.
CiscoUnrecognizedFRU	Indicates the Field Replaceable Unit (FRU) has a product identification that is not recognized.
CiscoVlanPortStatusChange	Generated by a device when the value of vlanTrunkPortDynamicStatus object has been

SNMP Trap Configurations Provided by NNMi, continued

Incident Configuration Name	Description
	changed.
CiscoWarmStart	Occurs when an Cisco agent is reconfigured.
HSRPStateChange	Sent when an HSRP interface transitions to or from an Active or Standby state in a particular HSRP Group.
IetfVrrpStateChange	Sent when a standard VRRP interface transitions to or from a Master State in a particular VRRP Group. This trap is used by the standard VRRP protocol. It corresponds to the vrrpTrapNewMaster trap name.
OSPFIfStateChange	Signifies that there has been a change in the state of a nonvirtual OSPF interface.
OSPFNbrStateChange	Signifies that there has been a change in the state of a nonvirtual OSPF neighbor.
OSPFVirtIfStateChange	Signifies that there has been a change in the state of an OSPF virtual interface.
Rc2kTemperature	Signifies the SNMPv2c entity acting as an SNMP agent, has detected the chassis is overheating.
RcAggLinkDown	(*NNMi Advanced*) Signifies the operational state of the Multi-Link Trunk (MLT) Aggregator changed from Up to Down. (Link Aggregation)
RcAggLinkUp	(*NNMi Advanced*) Signifies the operational state of the Multi-Link Trunk (MLT) Aggregator changed from Down to Up. (Link Aggregation)
RcChasFanDown	Signifies the SNMPv2c entity, acting as an SNMP agent, has detected that the rcChasFanOperStatus object for one of its power supply units is about to transition to the Down state.
RcChasFanUp	Signifies the SNMPv2c entity, acting as an SNMP agent, has detected that the rcChasFanOperStatus object for one of its power supply units is about to transition to the Up state.
RcChasPowerSupplyDown	Signifies the rcChasPowerSupplyOperStatus object for one of its power supply units is about to transition to the Down state.

SNMP Trap Configurations Provided by NNMi, continued

Incident Configuration Name	Description
RcChasPowerSupplyUp	Signifies the SNMPv2c entity, acting as an SNMP Agent, has detected that the rcChasPowerSupplyOperStatus object for one of its power supply units is about to transition to the Up state.
Rcn2kTemperature	Signifies that the SNMPv2c entity, acting as an SNMP agent, has detected the chassis is overheating.
RcnAggLinkDown	(*NNMi Advanced*) Signifies the operational state of the Multi-Link Trunk (MLT) Aggregator Link changed from Up to Down. (Link Aggregation)
RcnAggLinkUp	(*NNMi Advanced*) Signifies the operational state of the Multi-Link Trunk (MLT) Aggregator Interface has changed from Down to Up. (Link Aggregation)
RcnChasFanDown	Signifies the SNMPv2c entity, acting as an SNMP agent, has detected that the rcChasFanOperStatus object for one of its power supply units is about to transition into the Down state.
RcnChasFanUp	Signifies the SNMPv2c entity, acting as an SNMP agent, has detected that the rcChasFanOperStatus object for one of its power supply units is about to transition into the Up state.
RcnPowerSupplyDown	Signifies the SNMPv2c entity, acting as an SNMP agent, has detected that the rcChasPowerSupplyOperStatus object for one of its power supply units is about to transition into the Up state.
RcnPowerSupplyUp	Signifies the SNMPv2c entity, acting as an SNMP agent, has detected that the rcChasPowerSupplyOperStatus object for one of its power supply units is about to transition into the Up state.
RcnSmltIsLinkDown	Signifies the rcChasPowerSupplyOperStatus object for one of its power supply units is about to transition into the Down state.
RcnSmltIsLinkUp	Signifies the rcChasPowerSupplyOperStatus object for one of its power supply units is about

SNMP Trap Configurations Provided by NNMi, continued

Incident Configuration Name	Description
	to transition into the Up state.
RcSmltIsLinkDown	(*NNMi Advanced*) Signifies the operational state of the Split Multi-Link Trunk (SMLT) Aggregator Link is transitioning from Up to Down. (Link Aggregation)
RcSmltIsLinkUp	(*NNMi Advanced*) Signifies the operational state of the Split Multi-Link Trunk (SMLT) Aggregator Link is transitioning from Down to Up. (Link Aggregation)
RcVrrpStateChange	Sent when a Rapid City (RC) Nortel interface transitions to or from a Master state in a particular VRRP Group. This trap is used by the Rapid City (RC) Nortel proprietary VRRP protocol. It corresponds to the rcVrrpTrapNewMaster trap name.
RMONFallingAlarm	Sent when an RMON device falls below a preconfigured threshold.
RMONRiseAlarm	Sent when an RMON device exceeds a preconfigured threshold.
SNMPColdStart	Signifies that the sending protocol entity is reinitializing itself. Therefore, the agent's configuration or protocol might change.
SNMPLinkDown	Signifies that the sending protocol entity recognizes a failure in one of the communication links represented in the agent's configuration.
SNMPLinkUp	Signifies that the sending protocol entity recognizes that one of the communication links represented in the agent's configuration has come up.
SNMPWarmStart	Signifies that the sending protocol entity is reinitializing itself such that neither the agent configuration nor the protocol entity implementation is altered.
STPNewRoot	Indicates that the sending agent has become the new root of the Spanning Tree.
STPTopologyChange	Sent by a node when any of its configured ports transitions from the Learning state to the Forwarding state, or from the Forwarding state to the Blocking state.

NNMi Advanced. SNMP Trap Incident Configurations for HP Route Analytics Management Systems (RAMS)

Incident Configuration Name	Description
RexAdjStateDown	Signifies the adjacency went down.
RexAdjStateFlap	Signifies the adjacency's flap count (rexEventCount) in the duration given by rexCountDuration has become greater than or equal to rexEventThreshold. Both adjacency up and adjacency down count as flaps. For example: An adjacency going down and coming up increments the flap count by two.
RexAdjStateUp	Signifies the adjacency came up.
RexASPathChange	Signifies the AS path to a route has changed.
RexBgpRedundChange	Signifies a change in the number of next hops available for reaching a prefix
RexBgpVpnReachByCustGain	Signifies the routes in the Customer announced by Provider Edge (**PE**[1]) that are up and not baselined as compared to the threshold value. The deviation is represented as either of the following: • The number of routes in the Customer that are up and not baselined • The percentage of participating routes in the Customer that are up and not baselined
RexBgpVpnReachByCustLoss	Signifies the routes in the Customer announced by PE that are down and not baselined as compared to the threshold value. The deviation is represented as either of the following: • The number of routes in the Customer that are down and not baselined • The percentage of participating routes in the Customer that are down and not baselined
RexBgpVpnReachByRtGain	Signifies the routes in the Route Target announced by PE that are up and not baselined as compared to the threshold value. The deviation is represented as either of the following: • The number of routes in the Route Target that are up and not baselined • The percentage of participating routes in the Route

[1] Provider Edge router. The Internet Service Provider's router that receives your data on the path to your data's final desination. The Customer Edge (CE) router in your network connects to this PE.

NNMi Advanced. SNMP Trap Incident Configurations for HP Route Analytics Management Systems (RAMS), continued

Incident Configuration Name	Description
	Target that are up and not baselined
RexBgpVpnReachByRtLoss	Signifies the routes in the Route Target announced by PE that are down and not baselined as compared to the threshold value. The deviation is represented as either of the following: • The number of routes in the Route Target that are down and not baselined • The percentage of participating routes in the Route Target that are down and not baselined
RexPathChange	Indicates the a path attributes such as metric, number of hops, intermediate hops from a source router to a IP prefix or NSAP address have changed.
RexPeeringStateDown	Indicates a peering between a router and RAMS has gone down
RexPeeringStateFlap	Indicates a peering between a router and RAMS has gone down.
RexPeeringStateUp	Indicates a peering between a router and RAMS has come up.
RexPrefixDrought	Signifies a particular BGP Peer Rib has decreased significantly from the Baseline Size as a percentage of the baseline
RexPrefixFlood	Signifies a particular BGP Peer Rib has increased significantly from the Baseline Size as a percentage of the baseline.
RexPrefixStateDown	Indicates the prefix(rexDstPrfx,rexDstMask) announced by Router(rexSrcRtrSysID, rexSrcRtrIP, rexSrcRtrName) has gone down.
RexPrefixStateFlap	Indicates the prefix (rexDstPrfx,rexDstMask) flap count (rexEventCount) in the duration given by rexCountDuration becomes greater than or equal to rexEventThreshold. Both prefix up and prefix down count as flaps. For example: A prefix going down and coming up increments the flap count by two.
RexPrefixStateUp	Indicates the prefix(rexDstPrfx,rexDstMask) announced by Router(rexSrcRtrSysID, rexSrcRtrIP, rexSrcRtrName) has come up.

NNMi Advanced. SNMP Trap Incident Configurations for HP Route Analytics Management Systems (RAMS), continued

Incident Configuration Name	Description
RexRtrConnected	Indicates the first adjacency of a router becomes full duplex. This means the neighbor sends an LSA and the previously isolated router sends an LSA across that adjacency.
RexRtrIsolated	Signifies a router has become isolated from the rest of the topology as all of its duplex connections it has to other routers which are not overloaded with respect to a particular routing protocol have gone down.
RexRtrStateFlap	Signifies the router's flap count (rexEventCount) in the duration given by rexCountDuration has become greater than or equal to rexEventThreshold. Both router isolation and router connection count as flaps. For example: A router getting isolated and then connected increments the flap count by two.
RexTest	This trap is sent for test purposes
RexVpnPEParticipationByCustGain	Signifies the Provider Edges (PEs) participating in the Customer that are up and not baselined as compared to the threshold value. The deviation is represented as either of the following: • The number of PEs that are up and not baselined • The percentage of participating PEs that are up and not baselined
RexVpnPEParticipationByCustLoss	Signifies the Provider Edges (PEs) participating in the Customer that are down and not baselined as compared to the threshold value. The deviation is represented as either of the following: • The number of PEs that are down and not baselined • The percentage of participating PEs that are down and not baselined
RexBgpVpnReachByRtGain	Signifies the routes in the Route Target announced by PE that are up and not baselined as compared to the threshold value. The deviation is represented as either of the following: • The number of routes in the Route Target that are up and not baselined • The percentage of participating routes in the Route Target that are up and not baselined
RexVpnPEParticipationByRtLoss	Signifies the PEs participating in the Route Target (RT)

NNMi Advanced. SNMP Trap Incident Configurations for HP Route Analytics Management Systems (RAMS), continued

Incident Configuration Name	Description
	that are down and not baselined as compared to the threshold value. The deviation is represented as either of the following: - The number of PEs that are down and not baselined - The percentage of participating PEs that are down and not baselined
RexVpnReachByCustPEGain	Signifies the routes in the Customer announced by PE that are up and not baselined as compared to the threshold value. The deviation is represented as either of the following: - The number of routes in the Customer that are up and not baselined - The percentage of participating routes in the Customer that are up and not baselined
RexVpnReachByCustPELoss	Signifies the routes in the Customer announced by PE that are down and not baselined as compared to the threshold value. The deviation is represented as either of the following: - The number of routes in the Customer that are down and not baselined - The percentage of participating routes in the Customer that are down and not baselined
RexVpnReachByCustPrefixDown	Signifies that the prefix has become unreachable in Customer.
RexVpnReachByCustPrefixUp	Signifies that the prefix has become reachable in Customer.
RexVpnReachByRtPEGain	Signifies the routes in the Route Target announced by PE that are up and not baselined as compared to the threshold value. The deviation is represented as either of the following: - The number of routes in the Route Target that are up and not baselined - The percentage of participating routes in the Route Target that are up and not baselined
RexVpnReachByRtPELoss	Signifies the routes in the Route Target announced by PE that are down and not baselined as compared to the threshold value. The deviation is represented as either

NNMi Advanced. SNMP Trap Incident Configurations for HP Route Analytics Management Systems (RAMS), continued

Incident Configuration Name	Description
	of the following: • The number of routes in the Route Target that are down and not baselined • The percentage of participating routes in the Route Target that are down and not baselined
RexVpnReachByRtPrefixDown	Signifies the prefix has become unreachable in RT.
RexVpnReachByRtPrefixUp	Signifies that the prefix has become reachable in RT.
RexVpnSiteExpectedAnncdPfxLoss	Signifies that there is a decrease in the number of prefixes announced by the Vpn/Site pair.
RexVpnSiteExpectedRcvdPfxLoss	Signifies that there is a decrease in the number of prefixes received by the Vpn/Site pair.
RexVpnSitePrefixStateDown	Signifies the prefix(rexDstPrfx,rexDstMask) announced by Router(rexSrcRtrSysID, rexSrcRtrIP, rexSrcRtrName) in VPN(rexVpnName) and site (rexSiteName), has gone down.
RexVpnSitePrefixStateFlap	Signifies the prefix (rexDstPrfx,rexDstMask) flap count (rexEventCount) in the duration given by rexCountDuration becomes greater than or equal to rexEventThreshold. The prefix is announced by Router (rexSrcRtrSysID, rexSrcRtrIP, rexSrcRtrName) in VPN (rexVpnName) and site (rexSiteName). Both prefix up and prefix down count as flaps. For example: A prefix going down and coming up increments the flap count by two.
RexVpnSitePrefixStateUp	Signifies the prefix (rexDstPrfx,rexDstMask) has come up. The prefix is announced by Router (rexSrcRtrSysID, rexSrcRtrIP, rexSrcRtrName) in VPN (rexVpnName) and site (rexSiteName).
RexVpnSiteUnexpectedAnncdPfxGain	Signifies there is an increase in the number of prefixes announced by the Vpn/Site pair.
RexVpnSiteUnexpectedRcvdPfxGain	Signifies there is an increase in the number of prefixes received by the Vpn/Site pair.
TrafficHighLinkUtilization	Indicates the traffic volume has exceeded a specified threshold on a link. Specify the threshold as an

NNMi Advanced. SNMP Trap Incident Configurations for HP Route Analytics Management Systems (RAMS), continued

Incident Configuration Name	Description
	absolute number in kilobytes per second or in terms of percentage of link capacity.
TrafficLinkCoSUtilization	Indicates the traffic volume has exceeded a specified threshold for a CoS queue on a link. Specify the threshold as an absolute number in kilobytes per second or in terms of a percentage of link capacity.
TrafficLowLinkUtilization	Indicates the traffic volume has fallen below a specified threshold on a link. Specify the threshold as an absolute number in kilobytes per second or in terms of percentage of link capacity.
TrafficQuantityAlert	A generic trap for all non-link related traffic alerts. Specify the threshold as an absolute number in kilobytes per second or in terms of percentage of link capacity.

To see or modify these SNMP trap incident configurations:

1. Navigate to the **SNMP Trap Configuration** form.
   a. In the Workspace navigation panel, select the **Configuration** workspace.
   b. Select **SNMP Trap Configurations**
2. Select a row and click the Open icon.
3. When you finish, click **Save and Close**.

Syslog Message Incident Configurations Provided by NNMi

Caution: If a Syslog Message Incident configuration's **Author** value is **HP Network Node Manager**, it can be overwritten by NNMi. See Author form for important information.

NNMi provides the Syslog Message incident configurations described in the following tables. Each of the tables is organized by vendor.

You might also choose to create your own configurations for additional Syslog Messages that are important to you.

Syslog Message Configurations Provided by NNMi - CISCO

Syslog Message Configurations Provided by NNMi - CISCO

Incident Configuration Name	Description
BGP-5-ADJCHANGE	Indicates a Border Gateway Protocol (BGP) neighbor has either come

Syslog Message Configurations Provided by NNMi - CISCO, continued

Incident Configuration Name	Description
	up or gone down. This informational message normally appears as routers and BGP neighbors go up or down. However, unexpected neighbor loss might indicate high error rates or high packet loss in the network and should be investigated.
CDP-4-DUPLEX_ MISMATCH	Indicates that Cisco Discovery Protocol (CDP) has discovered a mismatch of duplex configuration. The recommended action is to configure the interfaces to the same duplex (full or half).
DTP-3-NONTRUNKPORTFAIL	Indicates that the port failed to become nontrunked.
DTP-3-TRUNKPORTFAIL	Indicates that the port failed to become trunked.
DTP-5-NONTRUNKPORTON	Indicates that the port is nontrunked.
DTP-5-TRUNKPORTCHG	Indicates that the encapsulation type of the trunk has changed.
DTP-5-TRUNKPORTON	Indicates that the port is trunked.
FR-5-DLCICHANGE	Indicates that a Frame-Relay Data Link Connection Identifier (DLCI) changes state. For states other than ACTIVE, such as INACTIVE and DELETED, check the Frame-Relay switch configuration to make sure its configuration matches the configuration of the router acting as the Frame-Relay Data Terminal Equipment device.
LINEPROTO-5-UPDOWN	Indicates the data link level line protocol changed state.
LINK-3-UPDOWN	Indicates the interface hardware went either up or down. The recommended action is to confirm the configuration settings for the interface, if the state change was unexpected.
LINK-4-ERROR	Indicates excessive errors have occurred on the interface. The recommended action is to check for duplex mismatches between both ends of the link.
OSPF-5-ADJCHG	Indicates an Open Shortest Path First (OSPF) neighbor has changed state.
PAGP-5-PORTFROMSTP	The switch has detected a loss of a link on a switch port, indicating the removal of a port from the spanning tree.
PAGP-5-PORTTOSTP	The switch has detected a link on a switch port, indicating the addition of a port to the spanning tree.

Syslog Message Configurations Provided by NNMi - CISCO, continued

Incident Configuration Name	Description
PORT_SECURITY-2-PSECURE_VIOLATION_VLAN	An unauthorized device attempted to connect on a secure trunk port.
SNMP-5-MODULETRAP	Indicates the SNMP agent has sent the Module Up or Module Down trap to the engine ID of the remote agent (or SNMP manager) because the corresponding module is up or down.
SPANTREE-5-PORTLISTEN	Indicates that the specified port in the VLAN state has changed to listening.
SPANTREE-5-ROOTCHANGE	Indicates that a new root port or a new root bridge has been selected for a specified spanning tree instance.
SPANTREE-6-PORTFWD	Indicates the port state in the VLAN changed to forwarding.
SPANTREE-6-PORTLISTEN	Indicates the port state in the VLAN changed to listening.
STACKMGR-6-MASTER_ELECTED	Indicates that the specified switch has been selected as the active switch.
STACKMGR-6-MASTER_READY	Indicates that the active switch is ready for use.
STACKMGR-6-STACK_LINK_CHANGE	Indicates that the status of the specified stack port has changed to active or inactive (up or down).
STANDBY-3-DUPADDR	Indicates that the router has received a Hot Standby Router Protocol (HSRP) message on the interface. The IP address in the HSRP message is the same as the IP address of the router. This condition may be caused by a network loop, a misconfiguration, or a malfunctioning switch.
STANDBY-6-STATECHANGE	Indicates that the Hot Standby Router Protocol (HSRP) state is changed.
SYS-3-MOD_CFGMISMATCH1	Indicates that a module was inserted into a slot that has been configured for another module type.
SYS-3-MOD_CFGMISMATCH2	Indicates that a module was inserted into a slot that has been configured for another module type.
SYS-3-MOD_CFGMISMATCH3	Indicates that a module was inserted into a slot that has been configured for another module type.
SYS-3-MOD_	Indicates that a module was inserted into a slot that has been

Syslog Message Configurations Provided by NNMi - CISCO, continued

Incident Configuration Name	Description
CFGMISMATCH4	configured for another module type.
SYS-3-PKTBUFBAD	Indicates that the packet buffer test detected a corrupted packet buffer on a module port.
SYS-3-PORT_COLL	Indicates that excessive or late collisions on the port are being logged.
SYS-3-PORT_ COLLDIS	Indicates that the threshold values for late or excessive collisions on a port have been exceeded.
SYS-3-PORT_IN_ ERRORS	Indicates that a port has experienced an input packet error.
SYS-3-PORT_RUNTS	Indicates that the switch has detected a runt frame (a frame that is less than 64 bytes). These errors are typically caused by physical layer issues or a speed/duplex mode mismatch with the remote device.
SYS-4-SYS_LCPERR4	Indicates a transient Application-Specific Integrated Circuit (ASIC) packet buffer problem.
SYS-5-MOD_INSERT	Indicates that the module was inserted.
SYS-5-MOD_OK	Indicates that the module passed diagnostic self-test and is online.
SYS-5-MOD_REMOVE	Indicates that module was removed.
SYS-5-MOD_RESET	Indicates that the system was reset from the specified console number or IP address.
SYS-5-RELOAD	Indicates that a reload was requested.
SYS-5-RESTART	Indicates that a restart was requested.
SYS-5-SYS_LCPERR5	Indicate an error or a significant condition for a specified port.

Syslog Message Configurations Provided by NNMi - HC3

Syslog Message Configurations Provided by NNMi - HC3

Incident Configuration Name	Description
ARP/3/ROUTECONFLICT	Indicates that the device returned a route conflict when an Address Resolution Protocol (ARP) entry was added to the device.
ARP/5/ARP_ DUPVRRPIP	Indicates that a virtual IP address in a Router Redundancy Group using the Virtual Router Redundancy Protocol (VRRP) conflict was detected.

Syslog Message Configurations Provided by NNMi - HC3, continued

Incident Configuration Name	Description
BFD/5/BFD_CHANGE_FSM	Indicates that the finite state machine (FSM) of a Brute Force Detection (BFD) session has been changed.
BGP/5/BGP_RECHED_THRESHOLD	Indicates that the warning threshold of prefixes that can be received from a peer or peer group has been reached.
CFM/5/CFM_SAVECONFIG_SUCCESSFULLY	Indicates that the save configuration was successful.
DEV/4/BOARD_LOADING	Indicates that the specified board is loading a file.
DEV/4/FAN_FAILED	Indicates that the fan failed to run.
DEV/4/FAN_RECOVERED	Indicates that the fan state changed from failed or absent to normal.
DEV/4/LOAD_FINISHED	Indicates that the board has finished loading a file.
DEV/4/POWER_ABSENT	Indicates that the power has been removed.
DEV/4/POWER_FAILED	Indicates that the power state changed to failed.
DEV/4/POWER_RECOVERED	Indicates that the power state changed from failed or absent to normal.
DEV/4/SYSTEM_REBOOT	Indicates that the system is rebooting.
DEVM/2/BOARD_STATE_FAULT	Indicates that an Input/Output or slave boards state changed to fault.
DEVM/2/POWER_FAILED	Indicates that the power state changed to failed.
DEVM/3/BOARD_REMOVED	Indicates that an Input/Output or slave board has been removed from a slot.
DEVM/3/RPS_ABSENT	Indicates that the redundant power system (RPS) is removed.
DEVM/5/POWER_RECOVERED	Indicates that the power state changed from failed or absent to **OK**.
DEVM/5/RPS_NORMAL	Indicates that the Redundant Power System (RPS) state changed to normal.
DEVM/5/SYSTEM_REBOOT	Indicates that the system is rebooting.

Syslog Message Configurations Provided by NNMi - HC3, continued

Incident Configuration Name	Description
LDP/5/LDP_SESSION_DOWN	Indicates that the sessions state changed to down.
MSTP/5/MSTP_BPDU_RECEIVE_EXPIRY	Indicates that a non-designated port did not receive a Bridge Protocol Data Unit (BPDU) within the `rcvdInfoWhile` interval, thus aging out the information of the port.
NTP/5/NTP_SOURCE_LOST	Indicates that there was a system synchronization source lost.
OPTMOD/3/TYPE_ERR	Indicates that the transceiver type is not supported by the port hardware.
OPTMOD/4/MODULE_IN	Indicates that the module on this port is plugged in to the interface.
OPTMOD/4/MODULE_OUT	Indicates that the module on this port is not plugged in.
OPTMOD/5/CHKSUM_ERR	Indicates that the checksum of transceiver information is bad.
OPTMOD/5/IO_ERR	Indicates that the transceiver information Input/Output failed.
OPTMOD/5/MOD_ALM_OFF	Indicates that a module fault is gone, and the module on the port has recovered to normal.
OPTMOD/5/MOD_ALM_ON	Indicates that a module not ready fault of the module is detected, and the module on the port has some fault.
OSPF/5/OSPF_NBR_CHG	Indicates that an important neighbor state change event has occurred.
OSPF/6/OSPF_LAST_NBR_DOWN	Indicates a record of the last Open Shortest Path First (OSPF) neighbor down event.
PIM/5/PIM_NBR_DOWN	Indicates that a Protocol Independent Multicast (PIM) neighbor state change to down.
STM/3/STM_LINK_STATUS_DOWN	Indicates that the link status of an Intelligent Resilient Framework (IRF) port is down.
STM/4/LINK_STATUS_CHANGE	Indicates that the link status of an Intelligent Resilient Framework (IRF) port changed to up or down.
STM/6/STM_LINK_STATUS_UP	Indicates that the link status of an Intelligent Resilient Framework (IRF) port is up.
VRRP/6/VRRP_STATUS_CHANGE	Indicates that the status of the virtual router has changed.

Syslog Message Configurations Provided by NNMi - HP Procurve

Syslog Message Configurations Provided by NNMi - HP Procurve

Incident Configuration Name	Description
ProCurve-RMON_BOOT_CRASH_RECORD0	Indicates that the specified management module was rebooted.
ProCurve-RMON_BOOT_CRASH_RECORD1	Indicates a text message was generated explaining the reasons for a system failure, which may include the type of failure (out of resources or bus error), task name, file name and line number (bus address).
ProCurve-RMON_BOOT_NO_CRASH_RECORD	Indicates that the specified management module failed without saving a failure record.
ProCurve-RMON_BOOT_SELFTEST_FAILURE	Indicates that the Self test failed while the switch was booting up.
ProCurve-RMON_CHASSIS_FAN_STATUS	A fan has failed or a failed fan is no longer failing. The fan state is indicated by failure or **OK**. The number of times that the fan failed is also displayed.
ProCurve-RMON_CHASSIS_HEARTBEAT_FAILURE	Indicates that communication with the specified slot was lost.
ProCurve-RMON_CHASSIS_POWER_STATUS	Indicates that the one of the following power conditions exists: • The Redundant Power-Supply (RPS) is failing • The RPS is operational but the main power supply is failing • A failed power supply is no longer failing. The state of the main or RPS power supply is indicated as failure or **OK**. The number of times that the power supply failed is also displayed.
ProCurve-RMON_LACP_DYNAMIC_TRUNK_OFF_LINE	Indicates that the trunk is now off-line.
ProCurve-RMON_LACP_DYNAMIC_TRUNK_ON_LINE	Indicates that the trunk is now online.
ProCurve-RMON_	Indicates that an error condition occurred on the specified trunk port and that the port is blocked.

Syslog Message Configurations Provided by NNMi - HP Procurve, continued

Incident Configuration Name	Description
LACP_ERROR_ CONDITION_ BLOCK	
ProCurve-RMON_ PMGR_PORT_UP	Indicates that the specified port is now online.
ProCurve-RMON_ POEMGR_ INTERNAL_50V_ FAULT	Indicates that the internal power supply has faulted or a faulted power supply is now operational. The power supply state is indicated as faulted or **OK**.
ProCurve-RMON_ POEMGR_PD_ DENIED_POWER	Indicates that there is insufficient power available to power the Powered Device (PD) on the port and the port does not have sufficient PoE priority to take power from another active PoE port.
ProCurve-RMON_ POEMGR_PD_ OVERCURRENT	Indicates that the Powered Device (PD) connected to the port has requested more than 15.4 watts of power. This may indicate a short-circuit or other problem in the PD.
ProCurve-RMON_ SSH_DISABLED	Indicates that the Secure Shell (SSH) server has been disabled.
ProCurve-RMON_ SSH_ENABLED	Indicates that the Secure Shell (SSH) server has been enabled.
ProCurve-RMON_ STP_NEW_ROOT	Indicates that the Spanning Tree Protocol (STP) root MAC address has changed for the specified STP priority level.

To see or modify these Syslog Message Incident configurations:

1. Navigate to the **Syslog Message Configurations** form.
   a. In the Workspace navigation panel, select the **Configuration** workspace.
   b. Expand the **Incidents** folder.
   c. Select **Syslog Message Configurations**
2. Select a row and click the Open icon.
3. When you finish, click **Save and Close**.

Remote NNM 6.x/7.x Event Configurations Provided by NNMi

Caution: If a Remote NNM 6.x/7.x Event configuration's **Author** value is **HP Network Node Manager**, it can be overwritten by NNMi. See Author form for important information.

NNMi provides the remote NNM 6.x or 7.x incident configurations described in the following table.

Remote NNMi Out-of-the-Box Incident Configurations

Incident Configuration Name	Description
OvStationNormal	Generated when the status of an NNM 6.x or 7.x management station changes to normal/up.
OvStationCritial	Generated when the status of an NNM 6.x or 7.x management station changes to down/critical.
OvNodeWarning	Generated when NNMi detects the status of a node has become up (some or all interfaces in the node are up).
OVNodeMajor	Generated when NNMi detects the status of a node has become up (some or all interfaces in the node are up).
OvNodeMarginal	Generated when NNMi detects the status of a node has become up (some or all interfaces in the node are up).
OvNodeUp	Generated when NNMi detects the status of a node has become up (some or all interfaces in the node are up).
OvNodeDown	Generated when NNMi detects the status of a node has become down (all interfaces in the node are down).
OvIfUp	Generated when NNMi detects the status of an interface has come up, normally by responding to an ICMP Echo (ping) request.
OvIfDown	Generated when NNMi detects the status of an interface has come up, normally by responding to an ICMP (ping) request.
OvMessage	Generated by a user to display a message in the incident browser.
OvIfIntermittent	Generated when NNMi detects the status of an interface has gone down and up multiple times.
OvApaAddressUp	Generated by the NNMi Causal Engine when it detects that the address is responding to polls.
OvApaIfUp	Generated by the NNMi Causal Engine when it detects that the interface is responding to polls.
OvApaNodeUp	Indicates a node's status went from Down to Up.
OvApaConnUp	Indicates a connection's status went from Down to Up.
OvApaAggPortUp	Indicates the OperStatus for the logical aggregate port connection is Up. (Link Aggregation)
OvApaAggPortDown	Indicates the OperStatus for the logical aggregate port connection is Down. (Link Aggregation)
OvApaAggPortDegraded	Indicates the OperStatus for one of the physical port

Remote NNMi Out-of-the-Box Incident Configurations, continued

Incident Configuration Name	Description
	connections in the aggregate connection is Down. (Link Aggregation)
OvApaAggPortConnUp	Indicates that an aggregate port connection between two nodes is responding to polls and no interfaces are down on either side of the connection. (Link Aggregation)
OvApaAggPortConnDown	Indicates an aggregate port connection between two nodes is not responding to polls and all interfaces might be down on both sides of the connection. (Link Aggregation)
OvApaAddressDown	Indicates a node's address status went from Up to Down.
OvApaIfDown	Iindicates a node's interface status went from Up to Down.
OvApaNodeDown	Indicates a node's status went from Up to Down.
OvApaConnDown	Indicates a connection's status went from Up to Down.
OvAPaIfIntermittent	Indicates an interface's status has gone Down and Upmultiple times.
OvApaAddressIntermittent	Indicates a node's address status has gone Down and Up multiple times.
OvApaConnIntermittent	Indicates a network's connection status has gone Down and Up multiple times.
OvApaNodeIntermittent	Indicates a node's status has gone Down and Up multiple times.
OvApaNodeSNMPNotResponding	Indicates an SNMP agent is not responding to queries.
OvApaAggPortNotDegraded	Indicates all of the physical port connections in the aggregate connection are Up. (Link Aggregation)
OvApaIfRemoved	Indicates an interface has been removed.
OvApaBoardUp	Indicates a node's board status has gone from Down to Up.
OvApaBoardDown	Indicates a node's board status has gone from Up to Down.
OvApaBoardRemoved	Indicates a node's board has been removed.

To see or modify these Remote NNM 6.x and 7.x trap incident configurations:

1. Navigate to the **Remote NNM 6.x and 7.x Event Configurations** view.
   a. In the workspace navigation panel, select the **Configuration** workspace.
   b. Expand the **Incidents** folder.

   c. Select **Remote NNM 6.x and 7.x Event Configurations**.

2. Click the Open icon in the row representing the configuration you want to see or edit.

3. When you finish, click **Save and Close**.

Management Event Configurations Provided by NNMi

> **Caution:** If a Management Event configuration's **Author** value is **HP Network Node Manager**, it can be overwritten by NNMi. See Author form for important information.

Deduplication is not configured for out-of-the-box management events. See "Correlate Duplicate Incidents (Deduplication Configuration)" on page 659 for information about how to configure deduplication.

NNMi provides the incident configurations for management events. Click here for more information.

To see or modify these management event incident configurations:

1. Navigate to the **Management Event Configurations** view.
   a. In the workspace navigation panel, select the **Configuration** workspace.
   b. Expand the **Incidents** folder.
   c. Select **Management Event Configurations**.
2. Double-click the row representing the configuration you want to see or modify:
   - Management Event Configurations Provided by NNMi
   - Additional Management Event Configurations (*NNM iSPI Performance for Metrics*)
3. When you finish, click **Save and Close**.

Management Event Configurations Provided by NNMi

Incident Configuration Name	Description
AddressNotResponding	Indicate an address is not responding to ICMP. Reasons an address might not respond include: • Its node is down • A device, such as a router, has been mis-configured so that some addresses cannot be reached
AggregatorDegraded	(*NNMi Advanced*) Indicates one or more (but not all) physical interfaces that are part of the Aggregator Interface are not operational. (Link Aggregation)
AggregatorDown	(*NNMi Advanced*) Indicates the

Management Event Configurations Provided by NNMi, continued

Incident Configuration Name	Description
	operational status of the Aggregator Interface is down (if monitored), or all of the corresponding physical interfaces are Down. (Link Aggregation)
AggregatorLinkDegraded	(*NNMi Advanced*) Indicates any Aggregation Member Interface is operationally down on either node, when there is a connection between two Aggregator Interfaces. (Link Aggregation)
AggregatorLinkDown	(*NNMi Advanced*) Indicates the Aggregator Interface on either side of an Aggregator Layer 2 Connection is down. (Link Aggregation)
BufferOutOfRangeOrMalfunctioning	Indicates the buffer pool is exhausted or cannot meet demand.
CardDisabled	Indicates that the card has been disabled by the device administrator.
CardDown	Indicates the card is not responding to polls.
CardRemoved	Indicates the card was removed from a device.
CardInserted	Indicates a card was inserted into a device.
CardUndeterminedState	Indicates the card reported a non-normal state for some unspecified reason.
ConnectionDown	Indicate that both (or all) ends of a connection are not responding to SNMP queries.
CpuOutOfRangeOrMalfunctioning	Indicates any of 5 second, 1 minute, or 5 minute utilization averages is too high.
CrgFailover	Indicates the primary card (for example, Card Active) has moved from one card to the other in a Card Redundancy Group. The Card Redundancy Group is routing packets properly.
CrgMultiplePrimary	Indicates NNMi has identified multiple primary cards (for example, Card Active) in the Card Redundancy Group. This

Management Event Configurations Provided by NNMi, continued

Incident Configuration Name	Description
	typically indicates the communication between the cards in the group is malfunctioning.
CrgNoPrimary	Indicates NNMi is unable to identify a primary card (for example, Card Active) in the Card Redundancy Group. This typically indicates one of the following: • One card, or both cards, are down • NNMi has identified only secondary cards (for example Standby cards) in the group • Communication between cards in the group is malfunctioning.
CrgNoSecondary	Indicates NNMi cannot identify a secondard card (for example Card Standby) in the Card Redundancy Group. This typically indicates the following: • One of the two cards in the group is down. • NNMi has identified the other card as primary (for example, Card Active). • The Card Redundancy Group is functioning properly
CustomPollCritical	Indicates that a Polling Instance associated with the Custom Poller Collection is in a Critical State.
CustomPolledInstanceOutOfRange	Indicates that a Custom Polled Instance has reached or exceeded a Comparison Map value or Threshold configured for the associated Custom Node Collection.
CustomPollMajor	Indicates that a Polling Instance associated with the Custom Poller Collection is in a Major State.
CustomPollMinor	Indicates that a Polling Instance associated with the Custom Poller Collection is in a Minor State.
CustomPollWarning	Indicates that a Polling Instance associated with the Custom Poller Collection is in a Warning State.

Management Event Configurations Provided by NNMi, continued

Incident Configuration Name	Description
DuplicateCorrelation	Provided as a template for configuring deduplication for an incident to specify which attribute values NNMi must match to verify that an incident is a duplicate **Note:** The DuplicateCorrelation incident configuration does not support Suppression, Enrichment or Dampening.
FanOutOfRangeOrMalfunctioning	Indicates the specified fan is not operating correctly.
ForwardIncidentRateExceeded	(*NNMi Advanced*) Indicates that the volume of messages entering a Regional Manager's Global Network Management message queue has exceeded the configured rate limits.
HostedObjectTrapStorm	Indicates the trap rate threshold for a hosted object has been exceeded.
InterfaceDisabled	Indicates the interface has been explicitly disabled by the device administrator.
InterfaceDown	Indicates that the interface is not responding to polls.
IpSubnetContainsIpWithNewMac	Indicates the MAC Address corresponding to a particular IP Address has changed. Possible causes include a duplicate IP Address on this subnet.
IslandGroupDown	Indicates all nodes in a group of Layer 2 connected nodes do not respond to monitoring polls (for example, ICMP or SNMP). These groups are automatically discovered and contain all of the nodes that can be connected through NNMi topology. Typically, these are groups on one side of a WAN (wide area network) connection.
LicenseExpired	Indicates that the expiration date has passed for an instant-on or temporary

Management Event Configurations Provided by NNMi, continued

Incident Configuration Name	Description
	NNMi license key. See "Extend a Licensed Capacity" on page 1575.
LicenseMismatch	Indicates that the licensed capacity for NNMi does not match the licensed capacity for one of the following products in your network environment: - An NNMi Integration Enablement - HP Network Node Manager iSPI Network Engineering Toolset Software (NNM iSPI NET) - HP Network Node Manager iSPI Performance for Metrics Software - HP Network Node Manager iSPI Performance for Traffic Software **Note:** The licensed capacity count is cumulative for each licensed product (across all installed license keys for that licensed product). See "Extend a Licensed Capacity" on page 1575.
LicenseNodeCountExceeded	Indicates that the number of discovered nodes exceeds the licensed capacity for managed node count. See "Extend a Licensed Capacity" on page 1575.
ManagementAddressICMPResponseTimeAbnormal	Indicates an abnormal Internet Control Message Protocol (ICMP) response time from the NNMi management server to the selected node. ICMP messages are typically used for diagnostic or routing purposes for determining whether a host or router could not be reached. The incident is generated when NNMi detects a higher than configured ICMP response time between the NNMi management server and the selected node.
ManagementAddressICMPResponseTimeHigh	Indicates a high Internet Control Message Protocol (ICMP) response time from the management server to the selected node. ICMP messages are typically used for

Management Event Configurations Provided by NNMi, continued

Incident Configuration Name	Description
	diagnostic or routing purposes for determining whether a host or router could not be reached. The incident is generated when NNMi detects a higher than configured ICMP response time between the NNMi management server and the selected node.
MemoryQueueIncidentRateExceeded	(*NNMi Advanced*) Indicates the rate at which NNMi forwards incidents to the Global Manager has exceeded the maximum allowed. NNMi no longer forwards incidents generated from SNMP traps.
MessageQueueSizeExceeded	Indicates one of the queues connecting the stages for the Event Pipeline is above the configured limits. NNMi determines queue size limits based on memory size.
ModifiedConnectionDown	Indicates a connection has been disconnected, moved, or both and is not responding to SNMP queries.
NnmClusterFailover	Indicates the NNMi cluster detected a failure of the active server. NNMi services were started on the standby server.
NnmClusterLostStandby	Indicates the NNMi cluster active server lost its communication to the standby server.
NnmClusterStartUp	Indicates the NNMi cluster was started in a state where no active server was already present. Therefore the server was started in the active state.
NnmClusterTransfer	Indicates the system administrator moved the active state from one server to another. The NNMi services will then start on the new active server.
NodeDeleted	Indicates that the specified node was deleted from the NNMi topology.
NodeDown	Indicates that the NNMi Causal Engine has determined the node is down based on the following analysis:

Management Event Configurations Provided by NNMi, continued

Incident Configuration Name	Description
	100% of the addresses assigned to this node are unreachable The SNMP agent installed on this machine is not responding NNMi is communicating with at least two of the neighboring devices. And at least two neighboring devices report problems with connectivity to this node.
NodeOrConnectionDown	Indicate a node is not responding to an ICMP or SNMP query. It also indicates that only one neighbor is down so that the NNMi Causal Engine cannot determine whether the node or the connection is down.
NonSNMPNodeUnresponsive	Indicates that a Non-SNMP node is unresponsive. Reasons for this include: 1) The node is down, or 2) An undiscovered device in the path between the node and the NNMi management server is down.
PowerSupplyOutOfRangeOrMalfunctioning	Indicates a power supply for the Source Node is not operating correctly.
RateCorrelation	Provided as a template to measure the number of incoming incidents within a defined time period. **Note:** The rateCorrelation incident configuration does not support Suppression, Enrichment, or Dampening.
RrgDegraded	This incident occurs only in Router Redundancy Groups with more than two routers. Indicates the following: • The Router Redundancy Group still has a primary and secondary device. • The remaining devices in the group are down or in an unexpected protocol-specific state. For example, in HSRP other devices should be in Listen

Management Event Configurations Provided by NNMi, continued

Incident Configuration Name	Description
	state. Typically, the protocol-specific communication between routers is malfunctioning. However, the group is routing packets properly.
RrgFailover	Indicates a primary role moved from one device to another in a Router Redundancy Group (for example, HSRP Active or VRRP Master). Reasons for this incident include one or more of the following: • A router or interface in the Router Redundancy Group has gone down. • A tracked object (interface or IP address) in the Router Redundancy Group has gone down. Even though a fail-over occurred, the group is routing packets properly.
RrgMultiplePrimary	Indicates that multiple primary devices are identified in a Router Redundancy Group (for example, HSRP Active or VRRP Master). Typically, the protocol-specific communication between routers in the group is malfunctioning.
RrgMultipleSecondary	Indicates that more than one secondary device is identified in a Router Redundancy Group (for example, HSRP Standby). **Note:** This incident applies only to Router Redundancy Groups that allow only one secondary member. Typically, the protocol-specific communication between routers in the group is malfunctioning. Typically, the protocol-specific communication between routers in the group is malfunctioning.

Management Event Configurations Provided by NNMi, continued

Incident Configuration Name	Description
RrgNoPrimary	Indicates that no primary device is identified in a Router Redundancy group (for example, HSRP Active or VRRP Master) . This incident typically indicates one of the following: • Too many routers are down. • Protocol-specific communication between routers in the group is malfunctioning.
RrgNoSecondary	Indicates that no secondary device is identified in a Router Redundancy Group (for example, HSRP Standby or VRRP Backup). This incident typically indicates the following: • Protocol-specific communication between routers in the group is malfunctioning. • The group is routing packets properly because a single primary device has been identified.
SNMPAgentNotResponding	The SNMP agent is not responding to SNMP queries on the selected Node.
SNMPTrapLimitCritical	Indicates the number of SNMP traps persisted in the NNMi database is approaching the maximum allowed limit. After the maximum allowed limit is reached, NNM no longer accepts SNMP traps until the number of SNMP traps within the database is reduced using the nnmtrimincidents.ovpl command.
SNMPTrapLimitMajor	Indicates the number of SNMP traps persisted in the NNMi database has reached or exceeded 95% of the maximum limit. After the maximum limit is reached, NNMi only accepts traps required for Causal Engine analysis until the number of SNMP traps within the database has been reduced using the

Management Event Configurations Provided by NNMi, continued

Incident Configuration Name	Description
	nnmtrimincidents.ovpl command.
SNMPTrapLimitWarning	Indicates the number of SNMP traps persisted in the NNMi database has reached or exceeded 90% of the maximum limit. After the maximum limit is reached, NNMi no longer accepts SNMP traps until the number of SNMP traps within the database is reduced using the nnmtrimincidents.ovpl command.
TemperatureOutOfRangeOrMalfunctioning	Indicates the specified temperature sensor on the Source Node is too hot or too cold.
TrapStorm	Indicates a trap storm has occurred.
VoltageOutOfRangeOrMalfunctioning	Indicates the specified voltage on one of the Source Node's power supplies is out of range.

(*NNM iSPI Performance for Metrics*) For network performance monitoring, the HP Network Node Manager iSPI Performance for Metrics Software provides additional management event configurations. Click here for more information.

The Node Compoent performance threshold events have a Category value of **Performance** a Family value of **Node Component**, and a Nature of **Root Cause**.

The Interface performance threshold events have a Category value of **Performance** a Family value of **Interface**, and a Nature of **Root Cause**.

Additional Management Event Configurations (*NNM iSPI Performance for Metrics*)

Incident Configuration Name	Description
BackplaneAbnormal	Indicates the backplane utilization is abnormal based on the computed baseline.
BackplaneOutOfRange	Indicates the backplane utilization has gone above or below a threshold setting.
BufferAbnormal	Indicates the buffer utilization is abnormal based on the computed baseline.
CpuAbnormal	Indicates the CPU utilization is abnormal based on the computed baseline for one of the following: • CPU 5 second utilization • CPU 1 minute utilization

Additional Management Event Configurations (NNM iSPI Performance for Metrics), continued

Incident Configuration Name	Description
	- CPU 5 minute utilization
DiskSpaceAbnormal	Indicates disk space utilization is abnormal based on the computed baseline.
DiskSpaceOutOfRange	Indicates disk space utilization has gone above or below a threshold setting.
InterfaceFCSLANErrorRateHigh	*Local Area Network*. Indicates a Frame Check Sequence (FCS) error rate on the interface has gone above a threshold setting. The error rate is based on the number of frames that were received with a bad checksum (CRC value). Possible causes include collisions at half-duplex, a duplex mismatch, bad hardware (NIC, cable, or port), or a connected device generating frames with bad FCS.
InterfaceFCSWLANErrorRateHigh	*Wireless Local Area Network*. Indicates a Frame Check Sequence (FCS) error rate on the interface has gone above a threshold setting. The error rate is based on the number of frames that were received with a bad checksum (CRC value). Possible causes include collisions at half-duplex, a duplex mismatch, bad hardware (NIC, cable, or port), or a connected device generating frames with bad FCS.
InterfaceInputDiscardRateHigh	Indicates the input discard rate on the interface has exceeded a threshold setting. This rate is based on the reported change in the number of input packets on the interface and the discarded packet count.
InterfaceInputErrorRateAbnormal	Indicates the input error rate on the interface is abnormal based on the computed baseline. This range is based on the reported change in the number of input packets on the interface and the packet error count. Possible causes include include bad packet checksums, incorrect header information, and small packets.
IntefaceInputErrorRateHigh	Indicates the input error rate on the interface *crossed* a High threshold setting. This rate is based on the reported change in the number of input packets on the interface and the packet error count.
InterfaceInputQueueDropsHigh	Indicates the number of input queue drops on the

Additional Management Event Configurations (NNM iSPI Performance for Metrics), continued

Incident Configuration Name	Description
	interface*crossed* a High threshold setting. This range is based on the number of packets dropped because of a full queue. Possible causes include that the number of packet buffers allocated to the interface is exhausted or has reached its maximum threshold.
InterfaceInputUtilizationAbnormal	Indicates the input utilization on the interface is abnormal based on the computed baseline. This range is based on the interface speed and the reported change in the number of input bytes on the interface.
InterfaceInputUtilizationHigh	Indicates the input utilization on the interface *crossed* a High threshold setting. This percentage is based on the interface speed and the reported change in the number of input bytes on the interface.
InterfaceInputUtilizationLow	Indicates the input utilization on the interface *crossed* a Low threshold setting. This percentage is based on the interface speed and the reported change in the number of input bytes on the interface.
InterfaceInputUtilizationNone	Indicates there is no input utilization on the interface. This value is based on the interface speed and the reported change in the number of input bytes on the interface.
InterfaceOutputDiscardRateHigh	Indicates the output discard rate on the interface *crossed* a High threshold setting. This rate is based on the reported change in the number of input packets on the interface and the discarded packet count.
InterfaceOutputErrorRateHigh	Indicates the output error rate on the interface *crossed* a High threshold setting. This rate is based on the reported change in the number of output packets on the interface and the packet error count.
InterfaceOutputQueueDropsHigh	Indicates the number of output queue drops on the interface *crossed* a High threshold setting. This number is based on the number of packets dropped because of a full queue. Possible causes include a congested interface.
InterfaceOutputUtilizationAbnormal	Indicates the output utilization on the interface is abnormal based on the computed baseline. This range is based on the interface speed, and the reported change

Additional Management Event Configurations (NNM iSPI Performance for Metrics), continued

Incident Configuration Name	Description
	in the number of output bytes on the interface.
InterfaceOutputUtilizationHigh	Indicates the output utilization on the interface *crossed* a High threshold setting. This percentage is based on the interface speed and the reported change in the number of output bytes on the interface.
InterfaceOutputUtilizationLow	Indicates the output utilization on the interface *crossed* a Low threshold setting. This percentage is based on the interface speed and the reported change in the number of output bytes on the interface.
InterfaceOutputUtilizationNone	Indicates there is no output utilization on the interface. This value is based on the interface speed and the reported change in the number of output bytes on the interface.
InterfacePerformanceCritical	Indicates the interface performance has reached a Critical severity.
InterfacePerformanceWarning	Indicates that the interface performance has reached a Warning severity.
MemoryOutOfRangeOrMalfunctioning	Indicates the Source Node's memory pool is exhausted or cannot meet the demand for use.
MemoryAbnormal	Indicates the memory utilization is abnormal based on the computed baseline.

Incident Pair (Pairwise) Configurations Provided by NNMi

NNMi provides the pairwise configurations described in the following table.

Pairwise Configurations Provided by NNMi

Name	Description
CiscoLinkDownUpPair	Cancels a CiscoLinkDown incident with a CiscoLinkUp incident on the same interface for the same SNMP agent address. This configuration is used for known Cisco devices.
CiscoModuleDownUpPair	Cancels a Cisco Module Down incident with a Cisco Module Up incident from the same module and SNMP agent address.
OvApaAddressDownUpPair	Cancels an NNM 6.x or 7.x Node Down event with a NNM 6.x or 7.x Node Up event from the same

Pairwise Configurations Provided by NNMi, continued

Name	Description
	SNMP agent address.
OvApaAggPortConnDownUpPair	(*NNMi Advanced*) Cancels an NNM 6.x or 7.x APA Aggregate Port Connection Down event with an NNM 6.x or 7.x APA Aggregate Port Connection Up event. (Link Aggregation)
OvApaAggPortDegradeNotDegradePair	(*NNMi Advanced*) Cancels an NNM 6.x or 7.x APA Aggregate Port Degraded event with an NNM 6.x or 7.x APA Aggregate Port Not Degraded event on the same interface for the same SNMP agent address. (Link Aggregation)
OvApaAggPortDownUpPair	(*NNMi Advanced*) Cancels of an NNM 6.x or 7.x APA Aggregate Port Down event with an NNM 6.x or 7.x APA Aggregate Port Up event on the same interface for the same SNMP agent address. (Link Aggregation)
OvApaBoardDownUpPair	Cancels an NNM 6.x or 7.x APA Board Down event with an NNM 6.x or 7.x APA Board Up event from the same SNMP agent address.
OvApaConnDownUpPair	Cancels an NNM 6.x or 7. x APA Address Down event with an NNM 6.x or 7.x APA Address Up event on the same address for the same SNMP agent address.
OvApaIfDownUpPair	Cancels an NNM 6.x or 7.x APA If Down event with an NNM 6.x or 7.x APA If Up event on the same interface for the same SNMP agent address.
OvApaNodeDownUpPair	Cancels an NNM 6.x or 7.x APA Node Down event with an NNM 6.x or 7.x APA Node Up event from the same SNMP agent address.
OvIfDownUpPair	Cancels an NNM 6.x or 7.x If Down event with an NNM 6.x or 7.x If Up event on the same interface for the same SNMP agent address.
OvNodeDownUpPair	Cancels an NNM 6.x or 7.x Node Down event with an NNM 6.x or 7.x Node Up event from the same SNMP agent address.
RcAggLinkDownUpPair	(*NNMi Advanced*) Cancels an RcAggLinkDown incident with an RcAggLinkUp incident from the same Multi-Link Trunk (MLT) Aggregator ID (from the MIB) and SNMP agent address. (Link Aggregation)
RcChasFanDownUpPair	Cancels an RcChasFanDown incident with an RcChasFanUp incident from the same fan ID (from

Pairwise Configurations Provided by NNMi, continued

Name	Description
	the MIB) and SNMP agent address.
RcChasPowerSupplyDownUpPair	Cancels an RcChasPowerSupplyDown incident with an RcChasPowerSupplyUp incident from the same power supply ID (from the MIB) and SNMP agent address.
RcSmltIsLinkDownUpPair	Cancels an RcSmltIstLinkDown incident with an RcSmltIstLinkUp incident from the same SNMP agent address.
RcnAggLinkDownUpPair	(*NNMi Advanced*) Cancels an RcnAggLinkDown incident with an RcnAggLinkUp incident from the same Multi-Link Trunk (MLT) Aggregator Link ID (from the MIB) and SNMP agent address. (Link Aggregation)
RcnChasFanDownUpPair	Cancels an RcnChasFanDown incident with an RcnChasFanUp incident from the same fan ID (from the MIB) and SNMP agent address.
RcnChasPowerSupplyDownUpPair	Cancels an RcnChasPowerSupplyDown incident with an RcnChasPowerSupplyUp incident from the same power supply ID (from the MIB) and SNMP agent address.
RcnSmltIsLinkDownUpPair	Cancels an RcnSmltIstLinkDown incident with an RcnSmltIstLinkUp incident from the same SNMP agent address.
SnmpLinkDownUpPair	Cancels an SNMPLinkDown incident with an SNMPLinkUp incident on the same interface for the same SNMP agent address.

To see or modify these incident pair configurations:

1. Navigate to the **Pairwise Configurations** view.
   a. In the workspace navigation panel, select the **Configuration** workspace.
   b. Expand the **Incidents** folder.
   c. Select **Pairwise Configurations**.
2. Double-click the row representing the configuration you want to see or modify.

   See "Pairwise Configuration Form (Correlate Pairs of Incidents)" on page 665 for more information.
3. When you are finished, click **Save and Close** to save your changes.

About Custom Incident Attributes for an Incident

The Custom Incident Attributes (CIAs) form enables you to specify additional CIAs to be saved with an incoming incident. You can then use this information to enhance the incident. For example, the information might be added to the incident message or used to customize a severity for a particular CIA value.

When creating a CIA for an incident configuration, you can specify any of the following values:

- Custom Attribute on the source node
- Custom Attribute on the interface (source object)

You also specify the Custom Incident Attribute name that will store this information.

You can provide the required information within the following contexts:

"Configure Custom Incident Attributes to Enrich an Incident Configuration (Interface Settings) (SNMP Trap Incidents)" on page 816

"Configure Custom Incident Attributes to Enrich an Incident Configuration (Interface Settings) (Remote NNM 6.x/7.x Events)" on page 1254

"Configure Custom Incident Attributes to Enrich an Incident Configuration (Interface Settings) (Management Events)" on page 1111

"Configure Custom Incident Attributes to Enrich an Incident Configuration (Interface Settings) (Syslog Message)(HP ArcSight)" on page 967

Custom Incident Attributes Provided by NNMi (Information for Administrators)

NNMi uses custom incident attributes to attach additional information to incidents.

A subset of CIAs is available for any particular incident. Any relevant CIAs are displayed in the Incident form, on the Custom Attributes tab. There are two categories of possible CIAs:

1. **Custom incident attributes**
   - Provided by NNMi
   - Provided for NNM iSPI Performance for Metrics
2. **SNMP trap varbinds**
   - Identified by the Abstract Syntax Notation value (ASN.1). Varbinds are defined in MIB files that you can load into NNMi. See "Load SNMP Trap Incident Configurations" on page 771.

The following tables explain the custom incident attributes provided by NNMi.

Custom Incident Attributes Provided by NNMi

Name	Description
cia.address	This attribute value is determined by the `com.hp.nnm.trapd.useUdpHeaderIpAddress`property

Custom Incident Attributes Provided by NNMi, continued

Name	Description

	defined in the following file:
	Windows:
	`%NNM_DATA%\shared\nnm\conf\props\nms-jboss.properties`
	UNIX:
	`$NNM_DATA/shared/nnm/conf/props/nms-jboss.properties`
	When `com.hp.nnm.trapd.useUdpHeaderIpAddress=true`, the cia.address value is the User Datagram Protocol (UDP) header IP Address.
	When `com.hp.nnm.trapd.useUdpHeaderIpAddress=false`, both the cia.address and cia.originaladdress values contain the SNMP Agent IP Address. The `com.hp.nnm.trapd.useUdpHeaderIpAddress` property is false by default.
	See the "Maintaining NNMi" chapter in the *HP Network Node Manager i Software Deployment Reference* for more information.

cia.originaladdress	This attribute value is determined by the `com.hp.nnm.trapd.useUdpHeaderIpAddress`property defined in the following file:
	Windows:
	`%NNM_DATA%\shared\nnm\conf\props\nms-jboss.properties`
	UNIX:
	`$NNM_DATA/shared/nnm/conf/props/nms-jboss.properties`
	This Custom Incident Attribute enables you to access both the User Datagram Protocol (UDP) header IP Address and the SNMP Agent IP Address of the managed device.
	When `com.hp.nnm.trapd.useUdpHeaderIpAddress=true`, cia.originaladdress is the value of the SNMP Agent IP Address and the cia.address value is the User Datagram Protocol (UDP) header IP Address.
	When

Custom Incident Attributes Provided by NNMi, continued

Name	Description
	`com.hp.nnm.trapd.useUdpHeaderIpAddress=false`, both cia.originaladdress and cia.address values contain the SNMP Agent IP Address. The `com.hp.nnm.trapd.useUdpHeaderIpAddress` property is false by default. See the "Maintaining NNMi" chapter in the *HP Network Node Manager i Software Deployment Reference* for more information.
cia.agentAddress	The IP Address that is stored in the SNMPv1 trap data for the SNMP Agent that generated the trap.
cia.custompoller.mibInstance	Instance number used to identify the row in the MIB table that contains the MIB value. **Tip**: You can use this CIA in the Message Format for a Custom Poller incident.
cia.custompoller.instanceDisplayValue	Value that results from the Instance Display Configuration. **Tip**: You can use this CIA in the Message Format for a Custom Poller incident. See "MIB Expressions Form (Custom Poller)" on page 431 for more information.
cia.custompoller.instanceFilterValue	The instance of the MIB Variable after the MIB Filter is applied to the nodes in the specified Node Group. **Tip**: You can use this CIA in the Message Format for a Custom Poller incident. The MIB Filter Variable is specified when configuring a Custom Poller Collection. The MIB Filter is specified when configuring a Custom Poller Policy for the collection. See "Create a Custom Poller Collection" on page 421and "Create a Policy" on page 449 for more information.
cia.cardsRemoved	Comma-separated list of removed card names used for formatting the **Card Removed** incident message.
cia.cardsInserted	Comma-separated list of the inserted card names used for formatting the **Card Inserted** incident message.
cia.cardsRemoved	Comma-separated list of removed card names used for formatting the **Card Removed** incident message.
cia.cardsInserted	Comma-separated list of the inserted card names used for formatting the **Card Inserted** incident message.

Custom Incident Attributes Provided by NNMi, continued

Name	Description
cia.custompoller.collection	The Name of the associated Custom Poller Collection.
cia.custompoller.lastValue	The last polled value that caused a state change which generated the incident.
cia.custompoller.policy	The Name of the associated Custom Poller Policy.
cia.custompoller.variable.description	The description of the MIB expression being polled.
cia.custompoller.variable.expression	The MIB expression that was collected and the computed value that caused the incident.
cia.custompoller.variable.name	The Name of the MIB expression variable that caused the incident.
cia.custompoller.state	The state of the Custom Polled Instance for this incident.
cia.incidentDurationMs	The time measured in milliseconds between when NNMi detected a problem with one or more network devices to the time the problem was resolved. Use this CIA to track the total time a particular object in the network was down or unavailable. **Note**: This CIA is used only when NNMi's Causal Engine has analyzed and Closed the incident. Any time an incident is closed manually (for example, by the network operator), NNMi does not include cia.incidentDuration.
cia.internalAddress	If *static* Network Address Translation (NAT) is part of your network management domain, and the NNMi management server is outside of that static NAT domain, the NNMi administrator can configure this attribute to show the internal IP address that is mapped to the external management address of the selected incident's Source Node. **Note**: The external management IP addresss (public address) must be mapped to this internal address (such as private IPv4 address) using the Overlapping IP Address Mapping Form. See "Overlapping Address Mapping Form" on page 192 for more information. For more information about Overlapping IP Addresses in an NNMi network see "Overlapping Address Mapping" on page 191.
cia.island.name	Name NNMi uses to identify the nodes contained in the island. NNMi administrators can use this cia value in Launch Actions to display the associated table view or topology map. To launch the associated topology map, use the following syntax for the Launch Action **Full URL** attribute value:

Custom Incident Attributes Provided by NNMi, continued

Name	Description

	`/nnm/launch?cmd=showNodeGroup&name=${cias [name=cia.island.name].value`
	To launch the associated table view, use the following syntax for the Launch Action **Full URL** attribute value:
	`/nnm/launch?cmd=showView&view=allNodesTableView&nodegroup= ${cias[name=cia.island.name].value}`
	See "Configure Launch Actions" on page 1422 and "Attributes per Object Type for Full URLs" on page 1426 for more information.

cia.island.numberOfNodes	Number of nodes contained in the island. Use this number to determine the effect of the associated Island Down incident. See Island Group Down for more information.
cia.reasonClosed	The Conclusion information identifying the reason NNMi changed the incident's Lifecycle State to Closed. For example, NNMi might include an Interface Up Conclusion as the reason an Interface Down incident was closed. **Note**: This CIA is used when NNMi's Causal Engine has analyzed and Closed the incident. Software that is integrated with NNMi might also provide values for cia.reasonClosed. Any time an incident is closed manually (for example, by the network operator), NNMi does not include cia.reasonClosed.
cia.remotemgr	Hostname or IP address of the (*NNMi Advanced - Global Network Management feature*) NNMi Regional Manager that is forwarding the event
cia.securityGroup.name	Name value for the Security Group. See "Configure Security Groups (Security Group Form)" on page 560 for more information. **Note**: This CIA does not appear if the node is assigned to the Default Security Group provided by NNMi.
cia.securityGroup.uuid	UUID value for the Security Group. See "Configure Security Groups (Security Group Form)" on page 560 for more information. **Note**: This CIA does not appear if the node is assigned to the Default Security Group provided by NNMi.
cia.snmpoid	SNMP trap object identifier.
cia.sourceNodeLongName	Fully qualified DNS name for the incident's Source Node.

Custom Incident Attributes Provided by NNMi, continued

Name	Description
cia.tenant.name	Name value for the Tenant. See "Use the Tenant Form" on page 196 for more information. **Note**: This CIA does not appear if the node is assigned to the Default Tenant provided by NNMi.
cia.tenant.uuid	UUID value for the Tenant. See "Use the Tenant Form" on page 196 for more information. **Note**: This CIA does not appear if the node is assigned to the Default Tenant provided by NNMi.
cia.timeIncidentDetectedMs	The timestamp in milliseconds when NNMi first detected the problem associated with an incident. **Note**: This CIA is used only when NNMi's Causal Engine has analyzed and Closed the incident. Any time an incident is closed manually (for example, by the network operator), NNMi does not include cia.timeIncidentDetected.
cia.timeIncidentResolvedMs	The time when NNMi determines the problem associated with the incident is resolved. **Note**: This CIA is used only when NNMi's Causal Engine has analyzed and Closed the incident. Any time an incident is closed manually (for example, by the network operator), NNMi does not include cia.timeIncidentResolved.

For network performance monitoring, additional custom incident attributes are provided for your use. Click here for more information.

Many incidents are candidates for these custom incident attributes:

Information about configuring thresholds is in the following topics:

- "Configure Threshold Monitoring for Node Groups" on page 402
- "Configure Threshold Monitoring for Interface Groups" on page 381
- "Configure Threshold Information for a Custom Poller Collection" on page 441

Custom Incident Attributes Provided for Thresholding

Name	Description
cia.thresholdParameter	The Monitored Attribute that is being measured in the threshold's configuration settings. For example, **Input Utilization**.
cia.thresholdLowerBound	The configured value that when *crossed* indicates a low threshold situation.
cia.thresholdUpperBound	The configured value that when *crossed* indicates a high

Custom Incident Attributes Provided for Thresholding, continued

Name	Description
	threshold situation.
cia.thresholdPreviousValue	Threshold results from the previous Polling Interval. For example, the threshold results might change from **Nominal** to **High**, based on a change in the `cia.thresholdMeasuredValue`. See Interface Form: Performance tab for a list of additional example Threshold result values.
cia.thresholdCurrentValue	Threshold results from the most recent Polling Interval. For example, **High**.
cia.thresholdMeasuredValue	The most recent value of the Measured Attribute being monitored according to this threshold's criteria settings. This measurement is the average of all measurements taken during the last polling interval (determined by the NNMi State Poller).
cia.thresholdMeasurementTime	The time at which the threshold was *crossed*. The time appears in ISO 8601 format.

These CIAs are used in a variety of ways:

- In SNMP trap configurations. See "Configure SNMP Trap Incidents" on page 782.
- In management events. See "Configure Management Events" on page 1078.
- In automatic actions. See "Configure an Action for an Incident" on page 748.
- In correlation configurations. See "Manage the Number of Incoming Incidents" below.
- In Launch Action definitions (access through the Actions menu). See "Control the NNMi Console Menus" on page 1414.

Manage the Number of Incoming Incidents

NNMi's Causal Engine reduces the number of incidents by extensively evaluating problems and determining the root cause for you, whenever possible.

To help simplify the diagnosis of network faults, you can configure NNMi to manage the number of incidents that are displayed. To do so, use any of the following methods:

- **Disable the Incident configuration**. In the **Basics** group of the SNMP Trap Configuration, Management Event Configuration or Remote NNM 6.x/7.x Configuration form, verify that **Enabled** □ is cleared for each configuration for which you do not want NNMi to generate an Incident.
- **Use NNMi's Management Mode feature to set the Management Mode of the network object to Not Managed or Out of Service**. NNMi discards any incoming traps if the trap

source is **Unmanaged**[1]. See "Stop or Start Managing an Object" on page 456 for more information.

- **Use the Monitoring Configuration to specify that you do not want NNMi to monitor the network object.** NNMi discards most incoming traps if the source object is not monitored. See "Configure NNMi Monitoring Behavior" on page 340 for more information.
- **Identify additional criteria for or relationships between incoming incidents**. When these criteria or relationships occur, NNMi modifies the flow of incidents by recognizing the criteria or patterns of incoming management events or SNMP traps and nesting related incidents as correlated children.

These strategies can dramatically reduce the number of incidents and improve the value of the incidents displayed. For example, instead of displaying an entire incident storm typically generated by equipment and link failures, use the deduplication configuration to specify only the most meaningful incidents, and correlate the rest as children. Then it is faster and easier to identify the network problem. See "Establish Criteria or Relationships for Incoming Incidents" below for more information.

Related Topics

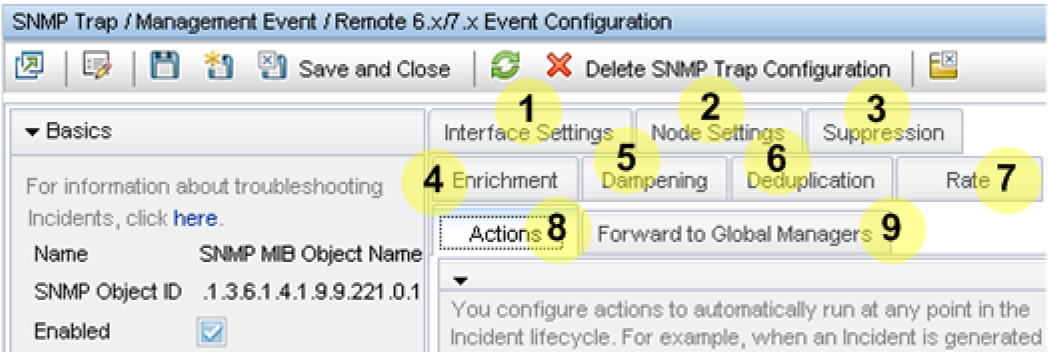
"Configure Management Events" on page 1078

"Configure SNMP Trap Incidents" on page 782

Establish Criteria or Relationships for Incoming Incidents

Using NNMi, you can establish the criteria or relationships for the incoming incidents using any of the incident configurations shown in the following diagram. You can choose to use them as is, edit them, or create your own configurations.

Incident Configuration Tabs


SNMP Trap / Management Event / Remote 6.x/7.x Event Configuration
Save and Close
Delete SNMP Trap Configuration
Basics
1 Interface Settings
2 Node Settings
3 Suppression
4 Enrichment
5 Dampening
6 Deduplication
Rate 7
Actions 8
Forward to Global Managers 9

For information about troubleshooting Incidents, click here.

Name SNMP MIB Object Name

SNMP Object ID .1.3.6.1.4.1.9.9.221.0.1

Enabled ☑

You configure actions to automatically run at any point in the Incident lifecycle. For example, when an Incident is generated

[1] Indicates the Management Mode is "Not Managed" or "Out of Service".

Click here for a description and example of each configuration option.

Overview of Incident Configuration Tabs

	Configuration Option	When to Use	Example
1	Interface Settings	Select this tab to specify that you want to configure Suppression, Enrichment, Dampening, and Actions for a specified Interface Group.	Change the Severity and Message of an incident configuration for a specified Interface Group. Dampen an Interface Down incident only for the interfaces in a specified Interface Group that you know will be intermittently unavailable.
2	Node Settings	Select this tab to specify that you want to configure Suppression, Enrichment, Dampening, Actions, and Diagnostic Selections for a specified Node Group.	Change the Severity and Message of an incident configuration for the nodes in a specified Node Group.
3	Suppression	Select this tab when you want to discard an incident before it appears in an incident view.	Discard an incident if it is in response to a particular status change notification trap.
4	Enrichment	Select this tab when you want to fine tune any of the following for a selected incident configuration: • Category • Family • Severity • Priority • Correlation Nature • Message • Assigned To	Change the Severity and Message of an incident configuration.

Overview of Incident Configuration Tabs, continued

	Configuration Option	When to Use	Example
		- Add a node or interface Custom Attribute to an incident	
5	Dampening	Select this tab to delay (dampen) the following for an incident configuration: - Appearance within Incident views in the NNMi Console - Execution of Incident Actions - Execution of Diagnostics (NNM iSPI NET)	Lengthen the Dampen Interval for the Interface Down incident Configuration provided by NNMi. Disable Dampening for the Interface Down Incident Configuration provided by NNMi.
6	Deduplication	Select this tab to correlate incidents that are identified as duplicates based on one or more Custom Incident Attribute (CIA) or SNMP trap varbind values. To help your operators understand the magnitude or significance of the problem, NNMi tracks the number of duplicates generated. This value is captured as the Duplicate Count attribute. It is incremented on the Duplicate Correlation incident. Its Correlation Nature attribute value is **Dedup Correlation**. NNMi also records the following information related to duplicate incidents: **First Occurrence Time**: Indicates the timestamp of the first occurrence of a duplicate incident. **Last Occurrence Time**: Indicates the timestamp of the latest notification for a set of duplicate incidents. **Count**: Specifies the number of duplicate incidents for the current configuration that NNMi stores at one time. For example, if the Count is 10, after NNMi receives 10 duplicate incidents, NNMi deletes the first (oldest) duplicate incident and keeps the eleventh. (NNMi stores ten maximum.) **Note**: A Duplicate Correlation incident inherits the Dampening settings of its Correlated Children. If the Correlated Children are Closed while Dampened, and therefore deleted, NNMi retains the parent Duplicate Correlation incident. See	Identify any CiscoLinkDown incidents as duplicates if the cia_address value is the same for the incident's Source Object.

Overview of Incident Configuration Tabs, continued

	Configuration Option	When to Use	Example
		"Dampening Incident Configurations" on page 679 for more information about Dampening an Incident Configuration.	
7	Rate	Select this tab to measure the rate of incoming incidents within a defined time period and correlate any incidents that occur within the specified time period. NNMi stores the following information related to rate: **Count**: Indicates the rate at which the incident must occur within the specified timeframe. **Hours, Minutes,** and **Seconds**: Used to measure the time within the rate must occur **First Occurrence Time**: Indicates the time at which the measured rate was reached. **Last Occurrence Time:** Indicates the last time which the incident occurred. NNMi updates the Correlation Notes with the number of incidents that have occurred within the specified time period. For example, 5 in 5 minutes. **Note**: A Rate Correlation incident inherits the Dampening settings of its Correlated Children. If the Correlated Children are Closed while Dampened, and therefore deleted, NNMi retains the parent Rate Correlation incident. See "Dampening Incident Configurations" on page 679 for more information about Dampening an Incident Configuration.	If a connection is intermittently down three times within 30 minutes; correlate the Connection Down incidents.
8	Actions	Select this tab to configure actions to automatically run at any point in the incident lifecycle. For example, you might want to configure an action to occur when an incident of the type you are configuring is generated (**Registered**).	When an incident is generated (**Registered**), open a trouble ticket. After the incident is **Closed**, close the trouble ticket.
9	Forward to Global Managers	*NNMi Advanced - Global Network Management feature*). Select the Global Manager Forwarding tab when you want to forward specific SNMP traps from your NNMi management server (a Regional Manager)	Forward all CiscoLinkDown incidents to the Global Manager.

Overview of Incident Configuration Tabs, continued

	Configuration Option	When to Use	Example
		to all Global Managers in your Global Network Management environment.	

You can also create Pairwise Configurations and your own Custom Correlations as described in the table below. See "About Pairwise Configurations" on page 660 and "Configure Custom Correlations" on page 680 for more information.

Additional Configuration Options

Configuration Option	When to Use	Example
Pairwise Configurations	Select the **Pairwise Configurations** view under the **Incidents** folder to pair the first occurrence of an incident to another subsequent incident. **Note**: NNMi provides Correlation Notes information only when the Causal Engine has analyzed and Closed the incident. Software that is integrated with NNMi might also provide information identifying the reason an incident was Closed. Any time an incident is Closed manually (for example, by the network operator), NNMi does not provide Correlation Notes information.	Correlate a CiscoLinkDown incident as the Child Incident for a CiscoLinkUp incident.
Custom Correlations	Select the **Custom Correlation Configuration** view under the **Incidents** folder of the **Configuration** workspace to correlate incidents using regular expressions to define the relationships between Parent and Child Incidents. This feature is useful when you want to define a relationship between a number of incidents potentially from different network objects that form a logical set to identify a problem. The set of correlations is considered complete if all of the incidents arrive within a specified time window. When configuring a Custom Correlation, you select the Parent and Child Incident configurations, the time window, and the regular expression that defines the relationship requirements that must be met before the incidents are correlated.	Correlate Interface Down incidents that occur for subinterfaces under the Interface Down incident generated for the main interface

See "Configuring Incidents" on page 589 for more information about the Incident Configuration options. See "Load SNMP Trap Incident Configurations" on page 771 for more information about how to specify which SNMP traps you want to receive by automatically creating or updating an incident configuration for an SNMP trap using a MIB file.

Related Topics

"Configure Management Events" on page 1078

"Configure SNMP Trap Incidents" on page 782

"Configure Remote NNM 6.x/7.x Events" on page 1221

Correlate Duplicate Incidents (Deduplication Configuration)

The deduplication configuration determines what values NNMi should match to detect when an SNMP trap, management event, or remote NNM 6.x/7.x event is a duplicate.

You can provide the required information within the following contexts:

"Deduplication Comparison Parameters Form (SNMP Trap Incident)" on page 907

"Deduplication Comparison Parameters Form (Syslog Message) (HP ArcSight)" on page 1057

"Deduplication Comparison Parameters Form (Remote NNM 6.x/7.x Events)" on page 1352

"Deduplication Comparison Parameters Form (Management Events)" on page 1200

Deduplication Comparison Parameters Form

Custom Incident Attributes (CIAs) are used as parameter values. Parameter values enable accurate identification of duplicate incidents. There are two categories of CIAs:

- SNMP trap varbind values (Name = the MIB varbind identifier, Type = asn_*)
- Custom attributes provided by NNMi (Name = cia.*, Type=String). See "Custom Incident Attributes Provided by NNMi (Information for Administrators)" on page 647.

You can provide the required information within the following contexts:

"Deduplication Comparison Parameters Form (SNMP Trap Incident)" on page 907

"Configure Deduplication for a Syslog Message Incident (HP ArcSight)" on page 1051

"Deduplication Comparison Parameters Form (Remote NNM 6.x/7.x Events)" on page 1352

"Deduplication Comparison Parameters Form (Management Events)" on page 1200

Track Incident Frequency (Rate: Time Period and Count)

Use Rate Configuration to track incident patterns *based on the number of incident reoccurrences within a specified time period*. After the count within the specified time period is reached, NNMi emits a Rate Correlation incident and continues to update the Correlation Notes with the number of occurrences within that rate.

You can provide the required information within the following contexts:

"Configure Rate (Time Period and Count) for an SNMP Trap Incident" on page 909

"Configure Rate (Time Period and Count) for a Syslog Message Incident (HP ArcSight)" on page 1059

"Configure Rate (Time Period and Count) for a Remote NNM 6.x/7.x Event Incident" on page 1353

"Configure Rate (Time Period and Count) for a Management Event Incident" on page 1202

About Pairwise Configurations

Often two incidents have a logical relationship to each other, for example, `CiscoLinkDown` followed by `CiscoLinkUp`. There is no need for both incidents to take up room in your Incident view. Nesting the two together helps you do your job quickly and efficiently.

Use the Pairwise Configuration to pair up the occurrence of one incident with another subsequent incident. When the second incident in the pair occurs, the first incident becomes a correlated child incident within the parent incident. See "Incident Pair (Pairwise) Configurations Provided by NNMi" on the next page for ideas.

When using Pairwise Configurations, note the following:

- You can use Payload Filters (for example, with trap varbinds) to identify the first and second incidents in a Pairwise Configuration.
- You can specify the same incident (for example, the same trap OID) as both the first and second incident configuration for a Pairwise Configuration.
- Using the Payload Filter to distinguish the first and second incidents (the first could represent a non-normal state and the second a normal state), different instances of the same incident configuration can cancel one another.
- You can also set up the Payload Filters such that the same incident instance cancels itself.
- You can use the same incident configuration in multiple Pairwise Configurations. For example:
  - Incident configuration A cancels both incident configuration B and incident configuration C
  - Incident configuration A cancels incident configuration B and incident configuration B cancels incident configuration C.
- A single incident instance can cancel multiple incident instances (for example, one Link Up trap cancels multiple instances of a Link Down trap)
- Use the Duration time to specify the time in which the second incident configuration cancels the first incident configuration. This Duration is calculated from the `originOccurrenceTime` of the second incident backwards in time, canceling any number of first incidents within the Duration specified.
- You can also specify whether to delete any incidents that were canceled according to the Pairwise Configuration and that occurred within the time period specified by the Duration attribute.
- When matching incidents, NNMi automatically takes into account the following values:
  - **SNMP Trap incidents**. NNMi takes into account from which device the trap originated using the `cia.address` value of the source address of the trap.
  - **Management Event incidents**. NNMi takes into account the name of the incident's Source Object and Source Node.

    Tip: NNMi displays the Name value used to identify the Source Node and Source Object in the **Source Node** and **Source Object** attribute for each incident in the Incident form.
  - **Remote NNM 6.x/7.x Event incidents**. NNMi uses the value of `cia.remotemgr` (IP Address or Hostname) of the NNM management station sending the incident and the

`cia.address` of the source address for the trap.

- **Syslog Message incidents**. NNMi does not automatically use any matching criteria.

Tip: When configuring the Matching Criteria, you do not need to specify any of the `cia` Names that NNMi automatically takes into account. See "Matching Criteria Configuration Form (Identify Incident Pairs)" on page 674 for more information.

Some incident pairs require extensive details to verify an accurate match. If both incidents have custom incident attributes, you can refine the match criteria beyond the values that NNMi automatically takes into account. See "Matching Criteria Configuration Form (Identify Incident Pairs)" on page 674 and "Configure a Payload Filter to Enrich a Pairwise Incident Configuration" on page 668 for more information.

Related Topics:

"Prerequisites for Pairwise Configurations" on page 664

"Pairwise Configuration Form (Correlate Pairs of Incidents)" on page 665

Incident Pair (Pairwise) Configurations Provided by NNMi

NNMi provides the pairwise configurations described in the following table.

Pairwise Configurations Provided by NNMi

Name	Description
CiscoLinkDownUpPair	Cancels a CiscoLinkDown incident with a CiscoLinkUp incident on the same interface for the same SNMP agent address. This configuration is used for known Cisco devices.
CiscoModuleDownUpPair	Cancels a Cisco Module Down incident with a Cisco Module Up incident from the same module and SNMP agent address.
OvApaAddressDownUpPair	Cancels an NNM 6.x or 7.x Node Down event with a NNM 6.x or 7.x Node Up event from the same SNMP agent address.
OvApaAggPortConnDownUpPair	(*NNMi Advanced*) Cancels an NNM 6.x or 7.x APA Aggregate Port Connection Down event with an NNM 6.x or 7.x APA Aggregate Port Connection Up event. (Link Aggregation)
OvApaAggPortDegradeNotDegradePair	(*NNMi Advanced*) Cancels an NNM 6.x or 7.x APA Aggregate Port Degraded event with an NNM 6.x or 7.x APA Aggregate Port Not Degraded event on the same interface for the same SNMP agent address. (Link Aggregation)
OvApaAggPortDownUpPair	(*NNMi Advanced*) Cancels of an NNM 6.x or 7.x APA Aggregate Port Down event with an NNM 6.x or 7.x

Pairwise Configurations Provided by NNMi, continued

Name	Description
	APA Aggregate Port Up event on the same interface for the same SNMP agent address. (Link Aggregation)
OvApaBoardDownUpPair	Cancels an NNM 6.x or 7.x APA Board Down event with an NNM 6.x or 7.x APA Board Up event from the same SNMP agent address.
OvApaConnDownUpPair	Cancels an NNM 6.x or 7. x APA Address Down event with an NNM 6.x or 7.x APA Address Up event on the same address for the same SNMP agent address.
OvApaIfDownUpPair	Cancels an NNM 6.x or 7.x APA If Down event with an NNM 6.x or 7.x APA If Up event on the same interface for the same SNMP agent address.
OvApaNodeDownUpPair	Cancels an NNM 6.x or 7.x APA Node Down event with an NNM 6.x or 7.x APA Node Up event from the same SNMP agent address.
OvIfDownUpPair	Cancels an NNM 6.x or 7.x If Down event with an NNM 6.x or 7.x If Up event on the same interface for the same SNMP agent address.
OvNodeDownUpPair	Cancels an NNM 6.x or 7.x Node Down event with an NNM 6.x or 7.x Node Up event from the same SNMP agent address.
RcAggLinkDownUpPair	(*NNMi Advanced*) Cancels an RcAggLinkDown incident with an RcAggLinkUp incident from the same Multi-Link Trunk (MLT) Aggregator ID (from the MIB) and SNMP agent address. (Link Aggregation)
RcChasFanDownUpPair	Cancels an RcChasFanDown incident with an RcChasFanUp incident from the same fan ID (from the MIB) and SNMP agent address.
RcChasPowerSupplyDownUpPair	Cancels an RcChasPowerSupplyDown incident with an RcChasPowerSupplyUp incident from the same power supply ID (from the MIB) and SNMP agent address.
RcSmltIsLinkDownUpPair	Cancels an RcSmltIstLinkDown incident with an RcSmltIstLinkUp incident from the same SNMP agent address.
RcnAggLinkDownUpPair	(*NNMi Advanced*) Cancels an RcnAggLinkDown incident with an RcnAggLinkUp incident from the same Multi-Link Trunk (MLT) Aggregator Link ID (from the MIB) and SNMP agent address. (Link

Pairwise Configurations Provided by NNMi, continued

Name	Description
	Aggregation)
RcnChasFanDownUpPair	Cancels an RcnChasFanDown incident with an RcnChasFanUp incident from the same fan ID (from the MIB) and SNMP agent address.
RcnChasPowerSupplyDownUpPair	Cancels an RcnChasPowerSupplyDown incident with an RcnChasPowerSupplyUp incident from the same power supply ID (from the MIB) and SNMP agent address.
RcnSmltIsLinkDownUpPair	Cancels an RcnSmltIstLinkDown incident with an RcnSmltIstLinkUp incident from the same SNMP agent address.
SnmpLinkDownUpPair	Cancels an SNMPLinkDown incident with an SNMPLinkUp incident on the same interface for the same SNMP agent address.

To see or modify these incident pair configurations:

1. Navigate to the **Pairwise Configurations** view.
   a. In the workspace navigation panel, select the **Configuration** workspace.
   b. Expand the **Incidents** folder.
   c. Select **Pairwise Configurations**.
2. Double-click the row representing the configuration you want to see or modify.

   See "Pairwise Configuration Form (Correlate Pairs of Incidents)" on page 665 for more information.
3. When you are finished, click **Save and Close** to save your changes.

Configure Pairwise Configurations

Use the Pairwise Configuration to pair the occurrence of one incident with another subsequent incident. See "About Pairwise Configurations" on page 660 for more information.

When configuring Pairwise Configurations you perform the following tasks:

- Use the Basics Pane of the Pairwise Configuration Form to Correlate Pairs of Incidents
- Optional. Configure the First and Second Incident Payload Filters
- Optional. Configure the Matching Criteria

Prerequisites for Pairwise Configurations

Tip: When configuring the Matching Criteria, you do not need to specify any of the ciaNames that NNMi automatically takes into account . See "Matching Criteria Configuration Form (Identify Incident Pairs)" on page 674 for more information.

When matching incidents, NNMi automatically takes into account the following values:

- **SNMP Trap incidents**. NNMi takes into account from which device the trap originated using the `cia.address` value of the source address of the trap.
- **Management Event incidents**. NNMi takes into account the name of the incident's Source Object and Source Node.

  Tip: NNMi displays the Name value used to identify the Source Node and Source Object in the **Source Node** and **Source Object** attribute for each incident in the Incident form.
- **Remote NNM 6.x/7.x Event incidents**. NNMi uses the value of `cia.remotemgr`.(IP Address or Hostname) of the NNM management station sending the incident and the `cia.address` of the source address for the trap.
- **Syslog Message incidents**. NNMi does not automatically use any matching criteria.

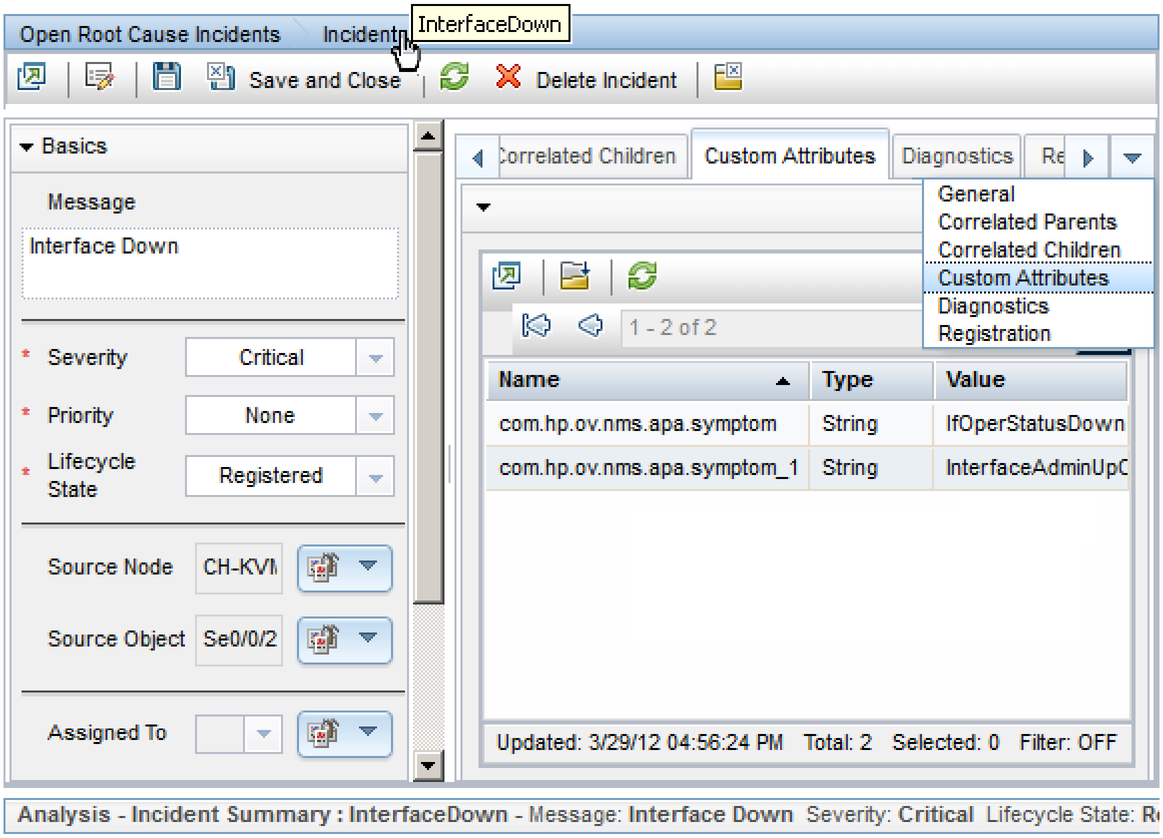
If you must provide more details to accurately identify the logical pair of incidents (from among all possible incidents related to that source node), complete the Optional step 6 below.

Complete the following steps before attempting to set up a Pairwise Configuration:

1. Identify the incidents or SNMP traps that consist of the logical relationship that makes the pair.

   Note: The incident configurations you select can be the same or different for each pair.
2. Configure those two incidents or traps within NNMi, if they are not already configured:
   - See "Incident Configurations Provided by NNMi" on page 605.
   - See "Configure SNMP Trap Incidents" on page 782.
   - See "Configure Remote NNM 6.x/7.x Events" on page 1221.
3. Generate one of each of the incidents or SNMP traps so you can see an example of each in one of the NNMi Incident views. See "Views Provided by NNMi".
4. To display the Incident form, double-click the row representing the first sample incident for the pair .

   Navigate to the Custom Attributes tab. These are the custom incident attributes available to use in step 6, below. See "Custom Incident Attributes Provided by NNMi (Information for Administrators)" on page 647 for more information about Custom Attributes.



Message: Interface Down

Severity: Critical

Priority: None

Lifecycle State: Registered

Source Node: CH-KVI

Source Object: Se0/0/2

Assigned To:

Name	Type	Value
com.hp.ov.nms.apa.symptom	String	IfOperStatusDown
com.hp.ov.nms.apa.symptom_1	String	InterfaceAdminUpC

Updated: 3/29/12 04:56:24 PM Total: 2 Selected: 0 Filter: OFF

Analysis - Incident Summary : InterfaceDown - Message: Interface Down Severity: Critical Lifecycle State: R

5. Repeat the previous step with the second sample incident for the pair.

6. *Optional*. If both Pairwise incidents have custom attributes, you can refine the match criteria beyond what NNMi automatically uses to determine a match. Some incident pairs require extensive details to verify an accurate match. See "Pairwise Configuration Form (Correlate Pairs of Incidents)" below.

Pairwise Configuration Form (Correlate Pairs of Incidents)

Use the Pairwise Configuration to pair the occurrence of one incident with another subsequent incident. See "About Pairwise Configurations" on page 660 for more information.

To configure incident pairs:

1. Complete the steps in "Prerequisites for Pairwise Configurations" on the previous page so you know exactly which two incidents or traps belong to this logical pair.

2. Navigate to the **Pairwise Configurations** form.

   a. From the workspace navigation panel, select the **Configuration** workspace.

   b. Expand the **Incidents** folder.

   c. Select **Pairwise Configurations**.

   d. Do one of the following:

   - To create a new pair configuration, click the ✱ New icon, and continue.
   - To edit an existing pair configuration, click the Open icon in the row representing the configuration you want to edit, and continue.
   - To delete a pair configuration, select a row and click the ✖ Delete icon.

3. Provide the basic definition of the pair of incidents for this correlation (see table).
4. When matching incidents, NNMi automatically takes into account the following values:
   - **SNMP Trap incidents**. NNMi takes into account from which device the trap originated using the `cia.address` value of the source address of the trap.
   - **Management Event incidents**. NNMi takes into account the name of the incident's Source Object and Source Node.

     Tip: NNMi displays the Name value used to identify the Source Node and Source Object in the **Source Node** and **Source Object** attribute for each incident in the Incident form.
   - **Remote NNM 6.x/7.x Event incidents**. NNMi uses the value of `cia.remotemgr`.(IP Address or Hostname) of the NNM management station sending the incident and the `cia.address` of the source address for the trap.
   - **Syslog Message incidents**. NNMi does not automatically use any matching criteria.

   Some incident pairs require additional details to verify an accurate match. If both incidents have custom incident attributes, you can refine the match criteria.

   Optional. Navigate to the **First Incident Payload Filter** and **Second Incident Payload Filter** tabs, and specify the payload filter to use when identifying a valid pair of incidents. See "Configure a Payload Filter to Enrich a Pairwise Incident Configuration" on page 668.

   Optional. Navigate to the **Matching Criteria** tab, and provide one or more custom incident attribute sets for NNMi to use as a filter when identifying a valid pair of incidents. See "Matching Criteria Configuration Form (Identify Incident Pairs)" on page 674.

   Tip: When configuring the Matching Criteria, you do not need to specify any of the ciaNames that NNMi automatically takes into account . See "Matching Criteria Configuration Form (Identify Incident Pairs)" on page 674for more information.

   Then, click **Save and Close** to save your changes and return to the previous configuration form.

The next time the two incidents in this pair are generated, the first one becomes a Child Incident of the second one. See "About Pairwise Configurations" on page 660 for an example.

Pairwise Configuration Definition

Attribute	Description
Name	The name is used to identify the pairwise configuration and must be unique. Use a name that will help you to remember the purpose for this pairwise configuration. Maximum length is 64 characters. Alpha-numeric characters are permitted. No spaces are permitted.
Enabled	In the **Basics** group, verify that **Enabled** ☑ is selected.

Pairwise Configuration Definition , continued

Attribute	Description
First Incident Configuration	Identify the incident in the pair that would occur first in the logical sequence. Click the ▾ Lookup icon and select **Quick Find**. Choose the name of one of the predefined incident configurations. If you cannot find it, see: • See "Incident Configurations Provided by NNMi" on page 605. • See "Configure SNMP Trap Incidents" on page 782. • See "Configure Remote NNM 6.x/7.x Events" on page 1221. This First Incident becomes the Child Incident when the Second (Parent) Incident occurs. For example, in the CiscoLinkDownUp Pairwise configuration, if a Cisco Link Up (Second Incident) occurs after a Cisco Link Down (First Incident), the Cisco Link Down is cancelled and correlated as a Child Incident under the Cisco Link Up.
Second Incident Configuration	Identify the incident in the pair that would occur second in the logical sequence.Click the ▾ Lookup icon and select **Quick Find**. Choose the name of one of the predefined incident configurations. If you cannot find it, see: • See "Incident Configurations Provided by NNMi" on page 605. • See "Configure SNMP Trap Incidents" on page 782. • See "Configure Remote NNM 6.x/7.x Events" on page 1221. This Second Incident becomes the Parent Incident if it occurs after the First Incident. For example, in the CiscoLinkDownUp Pairwise configuration, if a Cisco Link Up (Second Incident) occurs after a Cisco Link Down (First Incident), the Cisco Link Down is cancelled and correlated as a Child Incident under the Cisco Link Up.
Description	*Optional*. Explain the purpose of your pairwise configuration for future reference. Type a maximum of 1024 characters. Alpha-numeric, spaces, and special characters (~ ! @ # $ % ^ & * ( ) _+ -) are permitted.
Author	Indicates who created or last modified the Correlation Rule. **Caution:** If the Author attribute value is **HP Network Node Manager**, any changes are at risk of being overwritten in the future. • Click ▾ **Lookup** and select **Show Analysis** to display details about the currently selected Author. • Click **Quick Find** to access the list of existing Author values. • Click * **New** to create an Author value.
Duration	NNMi uses the value you enter to determine the duration window in which it correlates the Pairwise incidents you specify. During the timeframe specified, NNMi enables a single (parent) incident to cancel multiple (child) incidents.

Pairwise Configuration Definition , continued

Attribute	Description
	The Duration is calculated from the `originOccurrenceTime` of the parent incident backwards in time, canceling any child incidents within the Duration specified. Note the following: • By default, the Duration valus is 0 (zero). When the Duration value is 0, NNMi finds the most recently occurring incident that matches the First Incident specified in the Pairwise configuration, regardless of time. See First Incident Configuration for more information. • The maximum duration value is 365 days.
Delete when Canceled	When enabled ☑, after the Duration is reached, NNMi deletes any incidents that were canceled according to the Pairwise configuration and that occurred within the timeframe specified by the Duration attribute. When disabled, NNMi cancels the pairwise incidents as configured, but does not delete them.

Configure a Payload Filter to Enrich a Pairwise Incident Configuration

The Payload Filter Editor enables you to create expressions that further refine the filters used to select the incidents to be correlated in the Pairwise configuration. Make sure to design any complex Payload Filters offline as a Boolean expression first. This method can help to minimize errors when entering your expressions using the Payload Filter editor.

To create a Payload Filter expression for a Pairwise Incident configuration:

1. Navigate to the **Pairwise Configuration** form:
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Expand the **Incidents**folder.
   c. Select **Pairwise Configurations**.
   d. Do one of the following:
      i. To create an incident configuration, click the ✳ New icon, and continue.
      ii. To edit an incident configuration, select a row, click the 📂 Open icon, and continue.
      iii. To delete an incident configuration, select a row and click the ✖ Delete icon.
2. Select the **First Incident Payload Filter** or **Second Incident Payload Filter** tab.
3. Do one of the following:
   a. To create a new configuration, click the ✳ New icon.
   b. To edit an existing configuration, select a row, click the 📂 Open icon, and continue..

8. Define your Payload Filter (see table). Also see "Guidelines for Creating a Payload Filter". For more information about Payload Filters, click here.

   A Payload Filter enables you to further define the filters to be used for selecting the incidents to participate in the Pairwise Configuration. A Payload Filter selects incoming incidents based on Custom Incident Attribute names (ciaName) and values (ciaValue).

   a. Plan out the logic needed for your Filter String.

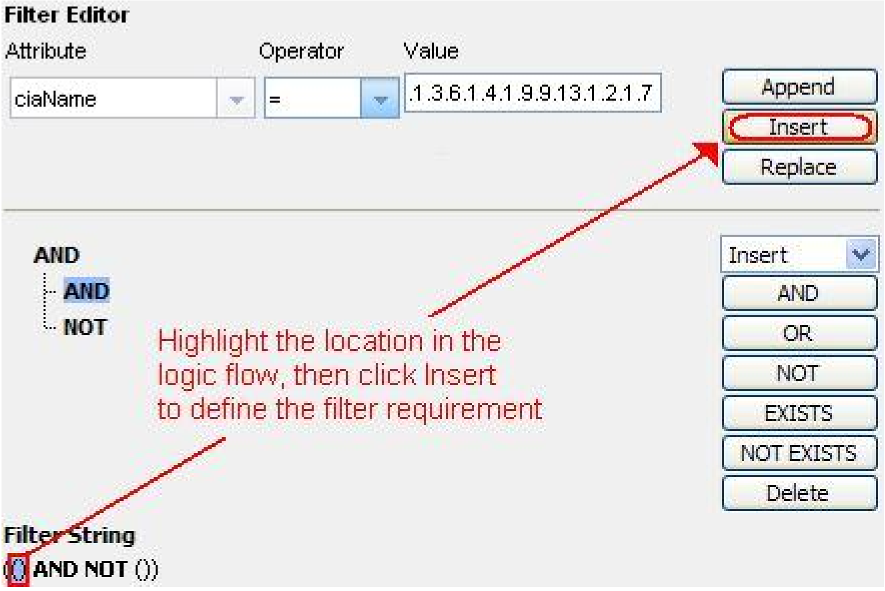
   b. Use the buttons on the bottom half of the Additional Filters Editor to establish the logic structure.

      For example, to establish the following structure, select **Insert**, then click **AND**, then **NOT**, and then **AND** a second time:

      ```
      (( ) AND NOT ( ))
      ```

   c. Now place your cursor in a location within the displayed Filter String, and use the top half of the filter editor to define the parameters of the highlighted filter requirement.

      For example, select a set of parentheses and use the Insert button to specify the filter requirement within those parentheses:



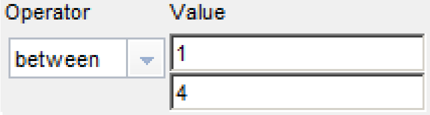
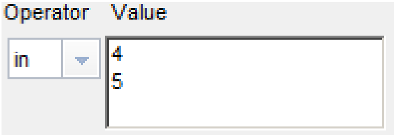
9. Click **Save and Close**.

10. Click **Save and Close** to save your changes and return to the previous form.

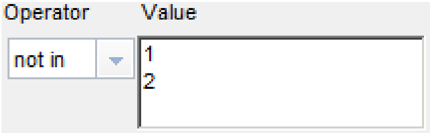
Payload Filter Editor Components

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: • ciaName • ciaValue
Operator	Valid operators are described below.

Payload Filter Editor Components, continued

Attribute	Description
	• **=** Finds all values equal to the value specified. Click here for an example. Example: `ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7` matches any incident that contains a varbind with the name value **.1.3.6.1.4.1.9.9.13.1.2.1.7**. • **!=** Finds all values not equal to the value specified. Click here for an example. Example: `ciaName != .1.3.6.1.4.1.9.9.13.1.2.1.7` matches any incident that contains a varbind with the name value other than **1.3.6.1.4.1.9.9.13.1.2.1.7**. • **<** Finds all values less than the value specified. Click here for an example. Example: `ciaValue < 6` matches any incident with a varbind value less than **6**. • **<=** Finds all values less than or equal to the value specified. Click here for an example. Example: `ciaValue <= 6` matches any incident with a varbind value less than or equal to **6**. • **>** Finds all values greater than the value specified. Click here for an example. Example: `ciaValue > 4` matches any incident with a varbind value greater than **4**. • **>=** Finds all values greater than or equal to the value specified. Click here for an example. Example: `ciaValue >= 4` matches any incident that contains a varbind with values greater than or equal to **4**. • **between** Finds all traps or events that include a varbind with a value equal to and between the two values specified. Click here for an example. Example: `ciaValue between`  Operator: between Value: 1 4 matches any incident that contains a varbind value equal to or greater than **1** and equal to or less than **4**. **Note**: As shown in the example, each value must be entered on a separate line. • **in** Finds any match to at least one value in a list of values. Click here for an example. Example: `ciaValue in`  Operator: in Value: 4 5

Payload Filter Editor Components, continued

Attribute	Description
	matches any incident with a varbind value of either **4** or **5**. **Note**: As shown in the example, each value must be entered on a separate line. NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example (**4, 5**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line. • **is not null** Finds all non-blank values. Click here for an example. Example: `ciaValue is not null` matches any incident with a varbind that contains a value. • **is null** Finds all blank values. Click here for an example. Example: `ciaValue is null` matches any incident with a varbind that does not have a value. • **like** Finds matches using wildcard characters. Click here for more information about using wildcard characters. The period asterisk (.*) characters mean *any number of characters of any type at this location*. The period (.) character means *any single character of any type at this location*. **Note**: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below. Examples: `ciaName like  \Q.1.3.6.1.4.1.9.9\E.*` finds all traps or events that contain varbind names that begin with **.1.3.6.1.4.1.9.9** and (optionally) end with any number of characters. `ciaValue like .*Chicago.*` finds all traps or events that contain a varbind value that includes the string **Chicago**. • **not between** Finds all values except those between the two values specified. Click here for an example. Example: `ciaValue not between 5 8` matches an incident that contains a varbind with the values less than **5** or greater than **8**. • **not in** Finds all values except those included in the list of values. Click here for an example. Example: `ciaValue not in`  Operator: not in Value: 1 2 matches any incident that contains a varbind with values other than **1** and **2**.

Payload Filter Editor Components, continued

Attribute	Description
	Note: As shown in the example, each value must be entered on a separate line. NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example, (**1, 2**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line. • **not like** Finds all that do not have the values specified (using wildcard strings). Click here for an example. The period asterisk (.*) characters mean *any number of characters of any type at this location*. The period (.) character means *any single character of any type at this location*. **Note**: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below. Example: `ciaName not like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that does not begin with **.1.3.6.1.4.1.9.9**. `ciaValue not like .*Chicago.*` finds all traps or events that do not contain a varbind value that includes the string **Chicago**.
Value	The value for which you want NNMi to search. Note the following: • The values you enter are case sensitive. • NNMi displays a variable number of value fields depending on the Operator selected. For example, the `between` Operator causes two value fields to be displayed. • The `between, in` and `not in` operators require that each value be entered on a separate line.

Additional Filters Editor Buttons

Button	Description
Append	Appends the current expression (Attribute, Operator,and Value) to the selected expression already included in the Filter String.
Insert	Inserts the current expression (Attribute, Operator,and Value) in front of the cursor location within the Filter String.
Replace	Replaces the selected expression with the expression displayed in the Attribute, Operator, and Value fields.
AND	Appends, inserts, or replaces the AND Boolean Operator in the selected cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the

Additional Filters Editor Buttons, continued

Button	Description
	expression as it is created.
OR	Appends, Inserts, or replaces the OR Boolean Operator in the current cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
NOT	Can be used in any part of the Filter String to specify that NNMi should exclude interfaces with values that pass the expression that immediately follows the `NOT`. For example, when evaluating the following expression, NNMi includes interfaces with (interface description) `ifDesc` containing **VLAN**, and excludes any Interfaces that have **VLAN10** for the (interface name) `ifName` value: `(ifDesc like VLAN AND NOT (ifName=VLAN10))` **Note**: View the expression displayed under Filter String to see the logic of the expression as it is created .
EXISTS	Used for filters that include Capabilities or Custom Attribute names and values in the Filer String. Indicates that you want NNMi to consider interfaces that do not have any Capabilities or Custom Attributes when evaluating the Filter String. **Note**: If you include Capabilities or Custom Attribute names and values in the Filer String, but do not use **EXISTS** or **NOT EXISTS**, NNMi excludes from its search interfaces that do not include Capabilities or Custom Attributes. For example, when evaluating the following Filter String, NNMi includes interfaces with (interface description) `ifDesc` containing **VLAN**, as well as any Interfaces Custom Attribute Role value is **LAN Connection to Oracle Server**: `(ifDesc like VLAN OR EXISTS((customAttrName=Role AND customAttrValue=LAN Connection to Oracle Server)))` **Note**: View the expression displayed under Filter String to see the logic of the expression as it is created.
NOT EXISTS	Used for filters that include Capabilities or Custom Attribute names and values in the Filer String. Indicates that you want NNMi to consider interfaces that do not have any Capabilities or Custom Attributes when evaluating the Filter String, but exclude the interfaces that match the expression that follows the **NOT EXISTS**. **Note**: If you include Capabilities or Custom Attribute names and values in the Filer String, but do not use **EXISTS** or **NOT EXISTS**, NNMi excludes from its search interfaces that do not include Capabilities or Custom Attributes. For example, when evaluating the following expression, NNMi includes interfaces with (interface description) `ifDesc` containing **VLAN**, and excludes any Interfaces that have the Custom Attribute Role and that Role value is **LAN**

Additional Filters Editor Buttons, continued

Button	Description
	Connection to Oracle Server: `(ifDesc like VLAN OR NOT EXISTS((customAttrName=Role AND customAttrValue=LAN Connection to Oracle Server)))` **Note**: View the expression displayed under Filter String to see the logic of the expression as it is created.
Delete	Deletes the selected expression. **Note**: If the Boolean Operator is selected, the Additional Filters Editor deletes all expressions associated with the Boolean Operator.

Matching Criteria Configuration Form (Identify Incident Pairs)

Tip: When configuring the Matching Criteria, you do not need to specify any of the ciaNames that NNMi automatically takes into account .

When matching incidents, NNMi automatically takes into account the following values:

- **SNMP Trap incidents**. NNMi takes into account from which device the trap originated using the `cia.address` value of the source address of the trap.
- **Management Event incidents**. NNMi takes into account the name of the incident's Source Object and Source Node.

  Tip: NNMi displays the Name value used to identify the Source Node and Source Object in the **Source Node** and **Source Object** attribute for each incident in the Incident form.
- **Remote NNM 6.x/7.x Event incidents**. NNMi uses the value of `cia.remotemgr`.(IP Address or Hostname) of the NNM management station sending the incident and the `cia.address` value of the source address for the trap.
- **Syslog Message incidents**. NNMi does not automatically use any matching criteria.

Some incident pairs require additional details to verify an accurate match. If both Pairwise incidents have custom incident attributes, you can use the Matching Criteria Configuration form to refine the matching criteria beyond what NNMi includes automatically.

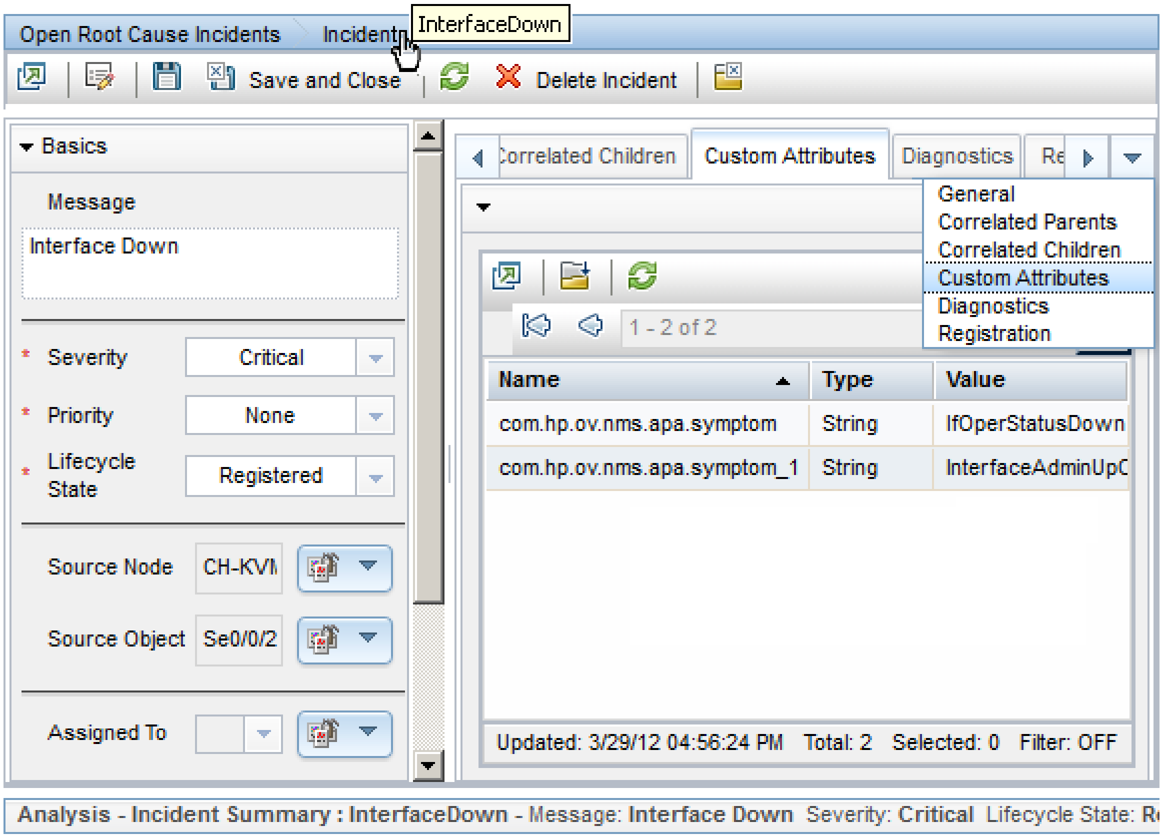
Tip: You can also use Payload Filters to define incident pairs. See "Configure a Payload Filter to Enrich a Pairwise Incident Configuration" on page 668for more information.

Specify one or more values for NNMi to use as a filter when identifying a valid pair of incidents.

You can use any Custom Incident Attributes (CIAs) displayed on the Incident form of the two incidents you are associating into a logical pair. The group of available CIAs depends on which incidents you select. There are two categories of possible CIAs:

- SNMP trap varbinds identified by the Abstract Syntax Notation value (ASN.1) or position. For example, a varbind OID of .1.3.6.1.2.1.2.2.1.1 or a position number of 25.
- Custom attributes provided by NNMi (Name = cia_*). See "Custom Incident Attributes Provided by NNMi (Information for Administrators)" on page 647.

The group of available CIAs depends on which incident you are configuring (for example, CiscoLinkDown). To see which CIAs are available, navigate to an Incident view, select an instance of that incident-type, click the Open icon and navigate to the Custom Attributes tab. The items listed in the table are the CIAs for that particular incident-type. For example, all CiscoLinkDown incidents would have the same group of CIAs shown in the illustration below.



Message: Interface Down

* Severity	Critical
* Priority	None
* Lifecycle State	Registered
Source Node	CH-KVI
Source Object	Se0/0/2
Assigned To	

Name	Type	Value
com.hp.ov.nms.apa.symptom	String	IfOperStatusDown
com.hp.ov.nms.apa.symptom_1	String	InterfaceAdminUpC

Updated: 3/29/12 04:56:24 PM Total: 2 Selected: 0 Filter: OFF

Analysis - Incident Summary : InterfaceDown - Message: Interface Down Severity: Critical Lifecycle State: R

For example:

- If you specify a First Incident Criteria and Second Incident Critieria of .1.3.6.1.2.1.2.2.1.1, the first incident's varbind value for the specified OID must match the second incident's varbind value for the specified OID to confirm a match.
- If you specify two custom attribute sets (one with both First Incident Criteria and Second Incident Criteria set to position 7, and one with both First Incident Criteria and Second Incident Criteria set to position 25), then the values for both custom attributes (varbind position 7 and varbind position 25) in both Incidents must match to confirm the logical pair.

To configure which attributes NNMi uses to verify incident identity:

1. Complete the steps in "Prerequisites for Pairwise Configurations" on page 664 so your choices for this Item Pair configuration are displayed in the NNMi console. (Two Incident forms should be open before you proceed to step 2.)
2. Navigate to the **Matching Criteria Configuration** form.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Expand the **Incidents** folder.

c. Select **Pairwise Configurations**.

d. Do one of the following:

- To create a new pairwise configuration, click the ✱ New icon.
- To edit a pairwise configuration, double-click the row representing the configuration you want to edit.

e. Navigate to the **Matching Criteria** tab.

f. Do one of the following:

i. To create a new matching criteria configuration, click the ✱ New icon.

ii. To edit a matching criteria configuration, double-click the row representing the configuration you want to edit.

3. Specify the Object Identifier (OID) or trap varbind position number you want NNMi to use to confirm the identity of the pair of incidents (see table).

4. Click **Save and Close** to save your changes and return to the previous form.

5. Repeat steps 1-3 any number of times. The incidents must pass all Matching Criteria, plus have identical Source Node and Source Object attribute values.

Matching Criteria Configuration

Attribute	Description
First Incident Criterion	Type the specification required to confirm the identify of the first incident in this logical pair of incidents. Provide one of the following: • The SNMP trap varbind Abstract Syntax Notation value - ASN.1 (OID) • The SNMP trap varbind position number **Caution**: The varbind position numbers are not visible in the table on the Incident form's Custom Attributes tab. And the rows in that table are sorted by the visible column headings and are not in varbind position order. You must access the vendor-supplied information in the underlying MIB file to determine the appropriate position number for any particular varbind. • The Custom Attribute **Name** value (see "Custom Incident Attributes Provided by NNMi (Information for Administrators)" on page 647 or the Name column in the table on the Incident Form: Custom Attributes Tab of the Incident you are configuring as a member of this logical pair).
Second Incident Criterion	Type the specification required to confirm the identify of the second incident in this logical pair of incidents. Provide one of the following: • The SNMP trap varbind Abstract Syntax Notation value - ASN.1 (OID) • The SNMP trap varbind position number **Caution**: The varbind position numbers are not visible in the table on the Incident form's Custom Attributes tab. And the rows in that table are sorted by the visible column headings and are not in varbind position order. You must access the vendor-supplied information in the underlying MIB file to

Matching Criteria Configuration , continued

Attribute	Description
	determine the appropriate position number for any particular varbind. • The Custom Attribute **Name** value (see "Custom Incident Attributes Provided by NNMi (Information for Administrators)" on page 647 or the Name column in the table on the Incident Form: Custom Attributes Tab of the Incident you are configuring as a member of this logical pair).

Related Topics

"Incident Pair (Pairwise) Configurations Provided by NNMi" on page 661

Pairwise Configuration Example

Tip: When configuring the Matching Criteria, you do not need to specify any of the ciaNames that NNMi automatically takes into account . See "Matching Criteria Configuration Form (Identify Incident Pairs)" on page 674 for more information.

When matching incidents, NNMi automatically takes into account the following values:

- **SNMP Trap incidents**. NNMi takes into account from which device the trap originated using the `cia.address` value.
- **Management Event incidents**. NNMi takes into account the unique name of the incident's Source Object and Source Node.

  Tip: NNMi displays the unique name value used to identify the Source Node and Source Object in the **Source Node** and **Source Object** attribute for each incident in the Incident form.
- **Remote NNM 6.x/7.x Event incidents**. NNMi uses the value of `cia.remotemgr`.
- **Syslog Message incidents**. NNMi takes into account the value of `event.deviceAddress`.

This example correlates the same ospfIfStateChange trap in a Pairwise Configuration. This example Pairwise Configuration, specifies that when the ospfIfState value changed from`1 (down)` to any value other than `1 (down)`, NNMi correlates the ospfIfStateChange incidents. See "Pairwise Configuration Form (Correlate Pairs of Incidents)" on page 665 for more information about how to specify a Pairwise configuration.

To use the same SNMP trap in a Pairwise configuration:

1. Navigate to the **Configuration** workspace.
2. Click to expand the **Incidents** folder.
3. Select **Pairwise Configurations**.
4. Click * (New) to create a Pairwise Configuration.
5. Enter a **Name** that is used to identify the Pairwise Configuration.
6. Make sure **Enabled** ☑ is checked.
7. In the **First Incident Configuration** attribute, select **Quick Find** from the ▾Lookup menu.

8. Select **OSPFIfStateChange**.
9. In the Second Incident Configuration attribute, select **Quick Find** from the ▾Lookup menu.
10. Select **OSPFIfStateChange**.
11. Enter a **Description** for the Pairwise Configuration.
12. Either leave the default value **Customer** or select **New** from the ▾ Lookup menu to specify an Author name.
13. Select **Days** from the **Duration** drop-down menu.
14. Enter the number of days in which NNMi correlates the Pairwise Configuration you specify .
15. If you want NNMi to delete the Pairwise incidents when they are canceled, click **Delete When Canceled**☑.
16. Navigate to the **First Incident Payload Filter** tab.
17. Use the following expression to indicate you want to use the OSPFIfState value of 1 (down):
    `ciaName = 1.3.6.1.2.1.14.7.1.12` AND `ciaValue = 1`
18. Navigate to the **Second Incident Payload Filter** tab.
19. Use the following expression to indicate you want to use any OSPFIfState value that is other than `1 (down):`
    `ciaName = 1.3.6.1.2.1.14.7.1.12` AND `ciaValue != 1`

    Note: You do not need to specify Matching Criteria. NNMi checks for a match using the value of `cia.address`.
20. Click **Save and Close** to save your changes and return to the Pairwise Configurations view.

Rate Comparison Parameters Form

Custom Incident Attributes (CIAs) are used as parameter values. Parameter values enable accurate identification of duplicate incidents. There are two categories of CIAs:

- SNMP trap varbind values (Name = the MIB varbind identifier, Type = asn_*)
- Custom attributes provided by NNMi (Name = cia.*, Type=String). See "Custom Incident Attributes Provided by NNMi (Information for Administrators)" on page 647.

You can provide the required information within the following contexts:

"Rate Comparison Parameters Form (SNMP Trap Incident)" on page 911

"Rate Comparison Parameters Form (Remote NNM 6.x/7.x Events)" on page 1355

"Rate Comparison Parameters Form (Management Events)" on page 1204

Suppress Incident Configurations

NNMi enables you to suppress incidents based on Interface Group, Node Group, or default Suppression settings. NNMi applies your Suppression settings in the following order. Only the first match applies.

1. Interface Group (SNMP Trap Configuration Form: Interface Settings tab)
2. Node Group (SNMP Trap Configuration Form: Node Settings tab)
3. Enrich configuration settings without specifying an Interface Group or Node Group (SNMP Trap Configuration Form: Enrichment tab)

You can provide the required information within the following contexts:

"Configure Suppression Settings for an SNMP Trap Incident" on page 879

"Configure Suppression Settings for a Management Event Incident" on page 1173

"Configure Suppression Settings for a Remote NNM 6.x/7.x Event Incident" on page 1317

Enrich Incident Configurations

NNMi enables you to fine tune and enhance incidents based on Interface Group, Node Group, or default Enrichment settings. NNMi applies your Enrichment settings in the following order. Only the first match applies:

1. Interface Group (Interface Settings tab)
2. Node Group (Node Settings tab)
3. Enrich configuration settings without specifying an Interface Group or Node Group (Enrichment tab)

The types of items you can fine tune and enhance for a selected incident configuration, include:

- Category
- Family
- Severity
- Priority
- Correlation Nature
- Message
- Assigned To

Note: Any configuration you specify for Severity, Priority, or Message overrides those values provided in the Basics information.

You can provide the required information within the following contexts:

"Configure Enrichment Settings for an SNMP Trap Incident" on page 887

"Configure Enrichment Settings for a Management Event Incident" on page 1181

"Configure Enrichment Settings for a Remote NNM 6.x/7.x Event Incident" on page 1325

Dampening Incident Configurations

NNMi enables you to delay (dampen) the following for an incident configuration:

- Appearance within Incident views in the NNMi Console
- Execution of Incident Actions
- Execution of Diagnostics (*HP Network Node Manager iSPI Network Engineering Toolset Software \ NNM iSPI NET*)

You can provide the required information within the following contexts:

"Configure Dampening Settings for an SNMP Trap Incident" on page 892

"Configure Dampening Settings for a Management Event Incident" on page 1185

"Configure Dampening Settings for a Remote NNM 6.x/7.x Event Incident" on page 1329

Configure Custom Correlations

For information about each Custom Correlation Configuration tab:

NNMi enables you to correlate groups of incidents under a Parent Incident. This feature is useful when you want to define a relationship between a number of incidents potentially from different network objects that form a logical set to identify a problem. The set of correlations is considered complete if all of the incidents arrive within a specified time window.

When configuring a Custom Correlation, you configure one or both of the following:

Rule	Description
Correlation Rule	**Tip**: Configure a Correlation Rule when you want to correlate only one type of Child incident Configuration with a Parent Incident Configuration that is generated by NNMi. Use a Correlation Rule to specify the following: • Parent Incident Configuration • Child Incident Configuration • Filters that NNMi should use when selecting the Parent and Child Incident instances for correlation • The time window within which NNMi begins to correlate the incidents. **Note**: If the Parent and Child incidents occur within the Correlation Window Duration specified, NNMi begins to correlate the incidents as soon as they occur. • The regular expression (Correlation Filter) that defines the relationship requirements that must be met before the incidents are correlated The Parent and Child Incident do not have to be the same incident configuration. For example, you can correlate an Address Not Responding incident with an Interface Down incident. See "Correlation Rule Example" on page 710 for a step-by-step example of how the Subinterface Correlation Rule provided by NNMi was created.

Rule	Description
Causal Rule	**Tip**: Configure a Causal Rule when you want to cause NNMi to generate a Parent Incident and you want to correlate one or more Child Incident Configurations under the Parent Incident that you cause to be generated. Use a Causal Rule to specify the following: • Parent Incident Configuration to be generated • One or more Child Incident Configurations to be correlated under the generated Parent Incident • Filters that NNMi should use when selecting the Child Incident instances for correlation • Source Object and Source Node filters to be used to determine the Source Node and Source Object for the generated Parent Incident • The time window that must be met before NNMi correlates the incidents. **Note**: NNMi waits until the Correlation Window Duration has passed before generating the Parent Incident and correlating its Child Incidents. To establish a relationship between multiple Custom Correlations, configure a Causal Rule to generate a Parent Incident that becomes the Child Incident of another Parent Incident. See "Causal Rule Example" for a step-by-step example of creating a Causal Rule.

To configure a Custom Correlation:

1. Navigate to the **Custom Correlation Configuration** form:
   a. From the workspace navigation pane, select the **Configuration** workspace.
   b. Expand the **Incidents** folder.
   c. Select **Custom Correlation Configuration**.
2. View the configured attributes (see table).
3. Do one of the following, or both:
   - Configure one or more Correlation Rules. See "Configure a Correlation Rule" on the next page for more information.
   - Configure one or more Causal Rules. See "Configure a Causal Rule" on page 714 for more information.
4. Click **Save and Close** to save your changes and return to the previous form.

Custom Correlation Registration Attribute

Attribute	Description
Last Modified	The date and time the Custom Correlation configuration was last modified.

Configure a Correlation Rule

Tip: Configure a Correlation Rule when you want to correlate a Child incident Configuration under a Parent Incident Configuration that is generated by NNMi.

Note: See **Help** → **Documentation Library** → **Release Notes**, and locate the **Support Matrix** link for Correlation Rule limitations.

When correlating groups of incidents under an existing Parent incident, use the Correlation Rules tab to specify the Correlation Rule that defines the Parent Incident, the Child Incident, and the relationship requirements that must be met before the incidents are correlated.

See "Correlation Rule Example" on page 710 for a step-by-step example of how the Subinterface Custom Correlation Rule provided by NNMi was created.

For information about each Correlation Rules tab:

To configure a Correlation Rule:

1. Navigate to the **Custom Correlation Configuration** form:
   a. From the workspace navigation pane, select the **Configuration** workspace.
   b. Expand the **Incidents** folder.
   c. Select **Custom Correlation Configuration**.
2. Navigate to the **Correlation Rules** tab.
3. From the **Correlation Rules** table toolbar, do one of the following:
   - To create a Correlation Rule, click the New icon, and continue.
   - To edit a Correlation Rule, click the Open icon in the row representing the Correlation Rule you want to edit, and continue.
   - To delete a Correlation Rule, click the Delete icon.
4. Create your Correlation Rule (see table).
5. Click **Save and Close** to save your changes and return to the previous form.

Correlation Rule Basic Attributes

Attribute	Description
Name	Type a maximum of 64 characters. Alpha-numeric, spaces, and special characters (~ ! @ # $ % ^ &amp; * ( ) _+ -) are permitted. The name is used to identify the Correlation Rule and must be unique. Use a name that will help you to remember the purpose of the Correlation Rule.
Author	Indicates who created or last modified the Correlation Rule. **Caution:** If the Author attribute value is **HP Network Node Manager**, any changes are at risk of being overwritten in the future.

Correlation Rule Basic Attributes, continued

Attribute	Description
	• Click **Lookup** and select **Show Analysis** to display details about the currently selected Author. • Click **Quick Find** to access the list of existing Author values. • Click **New** to create an Author value.
Enabled	If ☑ enabled, the NNMi Causal Engine uses the Correlation Rule when evaluating incidents. If □ disabled, the Correlation Rule is ignored.
Parent Incident	Specifies the incident configuration that should be used as the Parent Incident for the Correlation Rule. **Note**: If you want to create a rule to *generate* a Parent Incident configure a Causal Rule. See "Configure a Causal Rule" on page 714 for more information. **To specify a Parent Incident configuration**: 1. Click the Lookup icon, and do one of the following: ▪ To display Analysis Pane information, select Show Analysis. (See Use the Analysis Pane for more information about the Analysis Pane.) ▪ To display the list of possible incidents, select **Quick Find**. In the Quick Find dialog, select the Incident of interest. ▪ To create a Parent Incident, select one of the following: ○ New Management Event Configuration ○ New Remote NNM Event Configuration ○ New SNMP Trap Configuration ▪ To modify a Parent Incident, select Open. 2. *Optional*. To create or modify a Parent Incident, enter or modify the attribute values for the selected Incident configuration. See "Configuring Incidents" on page 589 for more information about the Incident Configuration form. 3. Click **Save and Close** to save your changes and return to the previous form.
Child Incident	Specifies the incident configuration that must match an incoming incident and that should be correlated as the Child Incident for the Custom Correlation. **To specify a Child Incident configuration**: 1. Click the Lookup icon, and do one of the following: ▪ To display Analysis Pane information, in the Quick Find dialog, select Show Analysis.(See Use the Analysis Pane for more information about the Analysis Pane.)

Correlation Rule Basic Attributes, continued

Attribute	Description
	▪ To create a Child Incident, select one of the following: ○ ✱ New Management Event Configuration ○ ✱ New Remote NNM Event Configuration ○ ✱ New SNMP Trap Configuration ▪ To modify a Child Incident, select Open. 2. *Optional*. To create or modify a Child Incident, enter or modify the attribute values for the selected Incident configuration. See "Configuring Incidents" on page 589 for more information about the Incident Configuration form. 3. Click **Save and Close** to save your changes and return to the previous form.
Correlation Window Duration	The time window within which NNMi begins to correlate the incidents. Enter a number for Days, Hours, Minutes, and Seconds. Note the following: • If the Parent and Child incidents occur within the Correlation Window Duration specified, NNMi begins to correlate the incidents as soon as they occur. • If you are relating multiple Custom Correlations, make sure the Correlation Window Duration allows enough time for all of the Parent and Child incidents to be generated. For example, when correlating a trap and an Interface Down incident on an interface that is polled every 5 minutes, use a 6-minute Correlation Duration Window to guarantee that the trap on the Interface Down occurs in the same Correlation Window Duration.This is because It might take up to 5 minutes for the associated Interfaced Down incident to occur **Note**: This example assumes that if the Interface Down occurs before the trap, the trap is sent within 6 minutes of the Interface Down Incident. • A lengthy Correlation Window Duration can increase memory usage and subsequently affect NNMi performance. When using a long duration window, the more often the incident occurs, the greater the affect on memory. To avoid possible performance issues, use a shorter duration for incidents that occur more frequently.
Description	Use the description field to provide additional information that you would like to store about the current incident configuration. This description applies only to the configuration entry. Type a maximum of 2048 characters. Alpha-numeric, spaces, and special characters (~ ! @ # $ % ^ & * ( ) _+) are permitted.

Configure a Parent Incident Filter for a Correlation Rule

Note: See **Help** → **Documentation Library** → **Release Notes**, and locate the **Support Matrix** link for Parent Incident Filter limitations.

Tip: The Parent Incident Filter is optional, but recommended. Use of a Parent Incident Filter improves NNMi performance by reducing the set of incidents that NNMi processes.

When correlating groups of incidents under a Parent Incident, you can define the requirements for the Parent Incident. The Parent Incident tab enables you to use the Filter Editor to define these requirements. For example, you might want to specify that the Source Node of the Parent Incident be a specific node Name pattern. See Valid Operators in the table that follows for examples of valid Parent Incident Filters.

When specifying the **like** or **not like** operator, use the syntax defined for Java regular expressions. For more information, see the Pattern (Java Platform SE6) API documentation at : `http://download.oracle.com/javase/6/docs/api/java/util/regex/Pattern.html`

Use the Filter Editor Buttons to insert Boolean Operators and to append, insert, and replace expressions in the Filter String. Use the Drag and Drop feature to make changes to the placement of the expressions in your Filter String. Click here for more information about using the Filter Editor for Custom Correlations:

- You can use Custom Incident Attributes, attributes for an incident's Source Node or Source Object, or both to define how matching incidents should be considered for the Correlation Rule. See Valid Attributes for more information.
- When specifying Attribute names and values, NNMi uses the type to determine a match. For example, if the Attribute type is numeric, NNMi does a numeric comparison. If the Attribute type is textual, NNMi does a lexographical string comparison. In all cases, when you use the **like** or **not like** operator, NNMi uses a lexographical string comparison. Click here for more information about Attribute types:
  - `ifIndex` and `ifSpeed` are numeric Attributes.
  - Any Attribute name that begins wtih "is" (`isSnmpInterface`, `isSnmpNode`, `isNnmSystemLocal`) represents a Boolean Attribute.
  - All other Attributes are textual.
- Each set of expressions associated with a Boolean Operator (for example, `AND`) is treated as if it were enclosed in parentheses and evaluated together.
- View the expression displayed under **Filter String** to see the logic of the expression as it is created.
- The `AND` and `OR` Boolean Operators must contain at least two expressions.
- The placement of your cursor and the subsequent text that is selected is important when performing operations using the Additional Filters Editor. For example, you append to or replace, the expression that is selected.
- The placement of your cursor and the subsequent text that is selected is especially important when adding your Boolean operators. See "Add Boolean Operators in the Additional Filters Editor" on page 310 for more information.

Filter Editor Buttons and Drag and Drop Feature

Button or Feature	Description
Append	Appends the current expression (Attribute, Operator,and Value) to the selected expression already included in the filter string.
Insert	Inserts the current expression (Attribute, Operator,and Value) in front of the cursor location within the Filter String.
Replace	Replaces the selected expression with the expression displayed Left or Right Expression.
AND	Inserts the AND Boolean Operator in the selected cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
OR	Inserts the OR Boolean Operator in the current cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
Delete	Deletes the selected expression. **Note**: If the Boolean Operator is selected, the Filter Editor deletes all expressions associated with the Boolean Operator.
Drag and Drop	You can drag any of the following items to a new location in the Filter String: ▪ Filter Editor Options: AND, OR, NOT, EXISTS, NOT EXISTS ▪ Filter Expression (Attribute, Operator and Value) When moving items in the Filter String, note the following: ▪ Click the item you want to move before dragging it to a new location. ▪ As you drag a selected item, an underline indicates the target location. ▪ If you are moving the selection up, NNMi places the item above the target location. ▪ If you are moving the selection down, NNMi places the item below the target location. ▪ If you attempt to move the selection to an invalid target location, NNMi displays an error message.

See "Correlation Rule Example" on page 710 for a step-by-step example of how the Subinterface Correlation Rule provided by NNMi was created.

To configure a Parent Incident Filter:

1. Navigate to the **Custom Correlation Configuration** form:

   a. From the workspace navigation pane, select the **Configuration** workspace.

   b. Expand the **Incidents** folder.

   c. Select **Custom Correlation Configuration**.

2. Navigate to the **Correlation Rules** tab.

3. From the **Correlation Rules** table toolbar, do one of the following:

   - To create a Correlation Rule, click the New icon, and continue.

   - To edit a Correlation Rule, click the Open icon in the row representing the configuration you want to edit, and continue.

   - To delete a Correlation Rule, click the Delete icon.

4. Navigate to the **Parent Incident Filter** tab.

5. Create your Parent Incident Filter (see Filter Editor Components).

   Filter Editor Components

Component	Description
Attribute	The Attribute on which NNMi searches. See Valid Attributes below for a description of valid Attributes.
Operator	Use this Operator to establish the relationship between the Attribute and Expression. See Valid Operators in the table below for the description of each valid Operator.
Expression	Use the Expression to complete the criteria for the Parent Incident configuration. See Valid Expressions below for a description of the components that you might include in your Expression.

6. Click **Save and Close** to save your changes and return to the previous form.

Valid Attributes

Attribute	Description
Attribute	The Attribute on which NNMi searches. Valid attributes other than Source Node attributes depend on the Incident's Source Object. NNMi checks the Source Node as well as the Source Object for any Capability value. Note the following when specifying Attributes: • Boolean Attributes begin with "is" and must contain the value `true` or `false`. • Use the following syntax to specify a Custom Incident Attribute (CIA): `valueOfCia(<CIA_Name>)` ▪ Check the appropriate Incident form for any valid CIA Names provided by NNMi. For example: `${valueOfCia(\Q.1.3.6.1.4.1.9.9.106.2.0.1\E)}= 5` ▪ When specifying the <*CIA_Name*>, you can use the syntax defined for Java regular expressions. For example, use the following syntax to specify that you

Valid Attributes , continued

Attribute	Description
	want to include any CIA Name that begins with .1.3.6.1.2.1.31.1.1.1.1.: `${valueOfCia(\Q.1.3.6.1.2.1.31.1.1.1.1.\E.*)}` ▪ Enclose all CIA names using the `\Q` and `\E` characters so that NNMi correctly interprets the period character. For example: `${child.valueOfCia(\Qcia.address\E)}` For more information, see the Pattern (Java Platform SE6) API documentation at `http://download.oracle.com/javase/6/docs/api/java/util/regex/Pattern.html` • If you use attributes that are valid for the Source Node, NNMi uses the Source Node when comparing values. If you use attributes that are valid for the Source Object, NNMi uses the Source Object when comparing values. You cannot use attributes that are valid for the Source Node and Source Object in the same filter. • When using attributes for a Source Object, note the following: ▪ When using attributes for a Source Object, the attribute must be valid for the incident's Source Object or NNMi does not find a match. For example, if you use the `hostedOn` attribute and the Source Object is not an interface, the correlation does not occur. **Tip:** To check a Source Object for an incident, select the incident of interest, then select Open from the Lookup menu for the Source Object, and examine the Source Object form. ▪ *SNMP Trap incidents only*. NNMi does not find a match when the value for a Source Object is `None`. A Source Object attribute value of `None` indicates that NNMi cannot resolve the Source Object. If you want to match the incident, use one or more Source Node attributes. • If the incident does not have a Source Object, a Source Node, or both (for example, the node is not stored in the NNMi database), you must use CIAs in your filter rather than Source Object or Source Node attributes. **Tip:** To check whether the Source Object or Source Node is stored in the NNMi database, open the incident and then select Open from the Lookup menu for the Source Node or Source Object displayed. If a form does not open for the selected object or node, this means the source is not stored in the NNMi database. • When specifying a Correlation Filter, precede the attribute name with either `parent.` or `child.` to specify from which incident the attribute value should be compared. For example, you might specify `${parent.hostedOn}` or `${child.ifDescr}`. Possible Source Object choices are as follows:

Valid Attributes , continued

Attribute	Description
	• Card [click here for a list of attribute values] **Unique Keys from the Card Form: Capabilities Tab**: ▪ capability (Unique Key of the Capability) • Interface [click here for a list of attribute values] Use the following syntax to specify a Custom Attribute (CA) for an Interface: `valueOfInterfaceCa(<`*`CA_Name`*`>)` For example: `${child.valueOfInterfaceCA(Role)} = WAN Connection` **Values from the Basics Attributes listed on the Interface Form**: ▪ hostedOn (Hosted On Node) You must use the full DNS name for the hostedOn value. **Values from the Interface Form: General Tab**: ▪ ifName (name configured for the interface) ▪ ifAlias (alias configured for the interface) ▪ ifDescr (description configured for the interface) ▪ ifIndex (index assigned to the interface) ▪ ifSpeed (speed configured for the interface) When entering the value for ifSpeed, use the actual numeric value for the interface speed. For example, use `10000000` for ifSpeed **10 Mbps**. **Addresses from the Interface Form: IP Addresses Tab**: ▪ ipAddress (IP Address associated with the interface) Because NNMi uses a lexicographical compare when evaluating IP addresses, it is recommended that you use the **like** and **not like** operators to specify IP address ranges rather than using the following operators: **>**, **>=**, **<**, or **<=**. **Unique Keys from the Interface Form: Capabilities Tab**: ▪ capability (Unique Key of the Capability) **Values from the Basics Attributes on the parent Node Form**: ▪ isSnmpInterface (Agent Enabled) **Values from the parent Node Form: General Tab**: ▪ sysOidInterface (System Object ID) **Values from the Basics Attributes on the associated Device Profile Form**: ▪ devVendorInterface (Device Vendor)

Valid Attributes , continued

Attribute	Description
	▪ devFamilyInterface (Device Family) • IP Address [click here for a list of attribute values] **Unique Keys from the IP Address Form: Capabilities Tab**: ▪ capability (Unique Key of the Capability) • Node [click here for a list of attribute values] Use the following syntax to specify a Custom Attribute (CA) for a Node: `valueOfNodeCa(<`*`CA_Name`*`>)` For example: `${valueOfNodeCa(Location)} = USA` **Values from the Basics Attributes on the Node Form**: ▪ hostname (Hostname, *case-sensitive*) ▪ mgmtIPAddress (Management Address) ▪ isSnmpNode (Agent Enabled) ▪ isNnmSystemLocal (NNMi Management Server) **Values from the Node Form: General Tab**: ▪ sysName (System Name) ▪ sysContact (System Contact) ▪ sysLocation (System Location) ▪ sysOidNode (System Object ID) **Addresses from the Node Form: IP Addresses Tab**: ▪ hostedIPAddress (Address) Because NNMi uses a lexicographical compare when evaluating IP addresses, it is recommended that you use the **like** and **not like** operators to specify IP address ranges rather than using the following operators: **>**, **>=**, **<**, or **<=** . **Unique Keys from the Node Form: Capabilities Tab**: ▪ capability (Unique Key of the Capability) **Values from the Basics Attributes on the associated Device Profile Form**: ▪ devVendorNode (Device Vendor) ▪ devFamillyNode (Device Family) **Values from the associated entry on the Regional Manager Form: Connection Tab**: ▪ nnmSystemName (Hostname, *case-sensitive*)

Valid Attributes , continued

Attribute	Description
	(*NNMi Advanced*) If the Global Network Management feature is enabled, this attribute value identifies a Regional Manager (NNMi management server). See "Global Manager: Connect to a Regional Manager" on page 103.

Valid Operator Values

Operator	Description

=	Finds all values equal to the value specified.
	Click here for examples.
	Match any incident with a CIA value of 5 and Object Identifier (OID) of .1.3.6.1.4.1.9.9.106.2.0.1:
	`${valueOfCia(\Q.1.3.6.1.4.1.9.9.106.2.0.1\E)} = 5`
	Match any incident with the Source Object's Capability equal to `com.hp.nnm.capability.card.fru`
	`$(capability) =  com.hp.nnm.capability.card.fru`

!=	Finds all values not equal to the value specified. Click here for an example. Match any incident with Device Vendor for the interface (Source Object) not equal to `Cisco`: `${devVendorInterface} != Cisco`
<	Finds all values less than the value specified. Click here for an example. Match any incident with a CIA value of less than 5 and Object Identifier (OID) of .1.3.6.1.4.1.9.9.106.2.0.1: `${valueOfCia(\Q.1.3.6.1.4.1.9.9.106.2.0.1\E)} < 5`
<=	Finds all values less than or equal to the value specified. Click here for examples. Match any incident with a CIA value of less than or equal to 5 and Object Identifier (OID) of .1.3.6.1.4.1.9.9.106.2.0.1: `${valueOfCia (\Q.1.3.6.1.4.1.9.9.106.2.0.1\E)} <= 5`
>	Finds all values greater than the value specified. Click here for an example. Match any incident with a CIA value of greater than 5 and Object Identifier (OID) of

Valid Operator Values, continued

Operator	Description
	.1.3.6.1.4.1.9.9.106.2.0.1: ${valueOfCia (\Q.1.3.6.1.4.1.9.9.106.2.0.1\E)} > 5
>=	Finds all values greater than or equal to the value specified. Click here for an example. Match any incident with a Source Object's (interface speed) ifSpeed value of 10Mbps: `${ifSpeed} >= 10000000`
is not null	Finds all non-blank values. Click here for an example. Match any incident with a Source Object's (interface name) ifName attribute that contains a value: `${ifName} is not null`
is null	Finds all blank values. Click here for an example. Match any incident with a Source Object's (interface name) ifName attribute that does not contain a value: `${ifName} is null`
like	Finds matches using the syntax defined for Java regular expressions. For more information, see the Pattern (Java Platform SE6) API documentation at: `http:/http://download.oracle.com/javase/6/docs/api/java/util/regex/Pattern.html` **Note:** To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E . The period asterisk (.*) characters mean *any number of characters of any type at this location*. The period (.) character means *any single character of any type at this location*. Click here for an example. Match any incident with a Source Object's (interface description) ifDescr attribute that includes `Serial` followed by one or more digits: `${ifDescr} like Serial\d+` Match any incident with a Source Object's (interface alias) ifAlias attribute that contains `EtherChannel` (for example, `PAgPEtherChannel Group 1`).

Valid Operator Values, continued

Operator	Description

	Note: The . (period) indicates any alphanumeric character.
	`${ifAlias} like .*EtherChannel.*`
	Match any incident with a CIA attribute value of `Chassis Fan Tray` followed by a digit and Object Identifier (OID) of `.1.3.6.1.4.1.9.9.13.1.4.1.3`
	Note: To include literal strings in the value, enclose the string value in `\Q<literal_value>\E` as shown in the following example.
	`${valueOfCia(\Q.1.3.6.1.4.1.9.9.13.1.4.1.3\E)} like Chassis Fan Tray \d`

not like	Finds all matches that do not have the values specified using the syntax defined for Java regular expressions. For more information,see the Pattern (Java Platform SE6) API documentation at : `http://download.oracle.com/javase/6/docs/api/java/util/regex/Pattern.html` **Note:** To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E . The period asterisk (.*) characters mean *any number of characters of any type at this location*. The period (.) character means *any single character of any type at this location*. Click here for an example. Match any incident with a Source Object's (interface name) ifName value that does not include `rtr`: `${ifName} not like .*rtr.*`

Valid Expressions

Attribute	Description
Expression	The value or pattern for which you want NNMi to search. Note the following: • The expression can include a valid Attribute. • The value or pattern you want to match is case sensitive. • When entering the value for ifSpeed, use the actual numeric value for the interface speed. For example, use `10000000` for ifSpeed **10 Mbps**.

Configure a Child Incident Filter for a Correlation Rule

> **Note:** See **Help → Documentation Library → Release Notes**, and locate the **Support Matrix** link for Child Incident Filter limitations.

> **Tip:** The Child Incident Filter is optional, but recommended. Use of a Child Incident Filter improves NNMi performance by reducing the set of incidents that NNMi processes.

When correlating groups of incidents under a Parent incident, you must specify the requirements for the Child Incident. The Child Incident tab enables you to use the Filter Editor to define these requirements. For example, you might want to specify that the Source Node of the Child Incident be a specific Node Name pattern. See Valid Operators in the table that follows for examples of valid Child Incident Filters.

Use the Filter Editor Buttons to insert Boolean Operators and to append, insert, and replace expressions in the Filter String. Use the Drag and Drop feature to make changes to the placement of the expressions in your Filter String. Click here for more information about using the Filter Editor for Custom Correlations:

- You can use Custom Incident Attributes, attributes for an incident's Source Node or Source Object, or both to define how matching incidents should be considered for the Correlation Rule. See Valid Attributes for more information.
- When specifying Attribute names and values, NNMi uses the type to determine a match. For example, if the Attribute type is numeric, NNMi does a numeric comparison. If the Attribute type is textual, NNMi does a lexographical string comparison. In all cases, when you use the **like** or **not like** operator, NNMi uses a lexographical string comparison. Click here for more information about Attribute types:
  - `ifIndex` and `ifSpeed` are numeric Attributes.
  - Any Attribute name that begins wtih "is" (`isSnmpInterface`, `isSnmpNode`, `isNnmSystemLocal`) represents a Boolean Attribute.
  - All other Attributes are textual.
- Each set of expressions associated with a Boolean Operator (for example, `AND`) is treated as if it were enclosed in parentheses and evaluated together.
- View the expression displayed under **Filter String** to see the logic of the expression as it is created.
- The `AND` and `OR` Boolean Operators must contain at least two expressions.
- The placement of your cursor and the subsequent text that is selected is important when performing operations using the Additional Filters Editor. For example, you append to or replace, the expression that is selected.
- The placement of your cursor and the subsequent text that is selected is especially important when adding your Boolean operators. See "Add Boolean Operators in the Additional Filters Editor" on page 310 for more information.

Filter Editor Buttons and Drag and Drop Feature

Button or Feature	Description
Append	Appends the current expression (Attribute, Operator,and Value) to the selected expression already included in the filter string.
Insert	Inserts the current expression (Attribute, Operator,and Value) in front of the cursor location within the Filter String.
Replace	Replaces the selected expression with the expression displayed Left or Right Expression.
AND	Inserts the AND Boolean Operator in the selected cursor location. **Note:** View the expression displayed under **Filter String** to see the logic of the expression as it is created.
OR	Inserts the OR Boolean Operator in the current cursor location. **Note:** View the expression displayed under **Filter String** to see the logic of the expression as it is created.
Delete	Deletes the selected expression. **Note:** If the Boolean Operator is selected, the Filter Editor deletes all expressions associated with the Boolean Operator.
Drag and Drop	You can drag any of the following items to a new location in the Filter String: ▪ Filter Editor Options: AND, OR, NOT, EXISTS, NOT EXISTS ▪ Filter Expression (Attribute, Operator and Value) When moving items in the Filter String, note the following: ▪ Click the item you want to move before dragging it to a new location. ▪ As you drag a selected item, an underline indicates the target location. ▪ If you are moving the selection up, NNMi places the item above the target location. ▪ If you are moving the selection down, NNMi places the item below the target location. ▪ If you attempt to move the selection to an invalid target location, NNMi displays an error message.

See "Correlation Rule Example" on page 710 for a step-by-step example of how the Subinterface Correlation Rule provided by NNMi was created.

To configure a Child Incident Filter:

1. Navigate to the **Custom Correlation Configuration** form:

    a. From the workspace navigation pane, select the **Configuration** workspace.

    b. Expand the **Incidents** folder.

    c. Select **Custom Correlation Configuration**.

2. Navigate to the **Correlation Rules** tab.

3. From the **Correlation Rules** table toolbar, do one of the following:

    - To create a Correlation Rule, click the New icon, and continue.
    - To edit a Correlation Rule, click the Open icon in the row representing the Correlation Rule you want to edit, and continue.
    - To delete a Correlation Rule, click the Delete icon.

4. Navigate to the **Child Incident Filter** tab.

5. Create your Child Incident Filter (see the Filter Editor Components below).

Filter Editor Components

Component	Description
Attribute	The Attribute on which NNMi searches. See Valid Attributes below for a description of valid Attributes.
Operator	Use this Operator to establish the relationship between the Attribute and Expression. See Valid Operators in the table below for the description of each valid Operator. **Note:** When specifying the **like** or **not like** operator, you must use the syntax defined for Java regular expressions. For more information, see the Pattern (Java Platform SE6) API documentation at: `http://download.oracle.com/javase/6/docs/api/java/util/regex/Pattern.html`
Expression	Use the Expression to complete the criteria for the Child Incident configurations. See Valid Expressions below for a description of the components that you might include in your Expression.

6. Click **Save and Close** to save your changes and return to the previous form.

Valid Attributes

Attribute	Description
Attribute	The Attribute on which NNMi searches. Valid attributes other than Source Node attributes depend on the Incident's Source Object. NNMi checks the Source Node as well as the Source Object for any Capability value.

Valid Attributes , continued

Attribute	Description
	Note the following when specifying Attributes: • Boolean Attributes begin with "is" and must contain the value `true` or `false`. • Use the following syntax to specify a Custom Incident Attribute (CIA): `valueOfCia(<CIA_Name>)` ▪ Check the appropriate Incident form for any valid CIA Names provided by NNMi. For example: `${valueOfCia(\Q.1.3.6.1.4.1.9.9.106.2.0.1\E)}= 5` ▪ When specifying the <*CIA_Name*>, you can use the syntax defined for Java regular expressions. For example, use the following syntax to specify that you want to include any CIA Name that begins with .1.3.6.1.2.1.31.1.1.1.1.: `${valueOfCia(\Q.1.3.6.1.2.1.31.1.1.1.1.\E.*)}` ▪ Enclose all CIA names using the `\Q` and `\E` characters so that NNMi correctly interprets the period character. For example: `${child.valueOfCia(\Qcia.address\E)}` For more information, see the Pattern (Java Platform SE6) API documentation at `http://download.oracle.com/javase/6/docs/api/java/util/regex/Pattern.html` • If you use attributes that are valid for the Source Node, NNMi uses the Source Node when comparing values. If you use attributes that are valid for the Source Object, NNMi uses the Source Object when comparing values. You cannot use attributes that are valid for the Source Node and Source Object in the same filter. • When using attributes for a Source Object, note the following: ▪ When using attributes for a Source Object, the attribute must be valid for the incident's Source Object or NNMi does not find a match. For example, if you use the `hostedOn` attribute and the Source Object is not an interface, the correlation does not occur. **Tip:** To check a Source Object for an incident, select the incident of interest, then select Open from the Lookup menu for the Source Object, and examine the Source Object form. ▪ *SNMP Trap incidents only*. NNMi does not find a match when the value for a Source Object is `None`. A Source Object attribute value of `None` indicates that NNMi cannot resolve the Source Object. If you want to match the incident, use one or more Source Node attributes. • If the incident does not have a Source Object, a Source Node, or both (for example, the node is not stored in the NNMi database), you must use CIAs in your filter rather than Source Object or Source Node attributes.

Valid Attributes , continued

Attribute	Description
	Tip: To check whether the Source Object or Source Node is stored in the NNMi database, open the incident and then select Open from the Lookup menu for the Source Node or Source Object displayed. If a form does not open for the selected object or node, this means the source is not stored in the NNMi database. • When specifying a Correlation Filter, precede the attribute name with either `parent.` or `child.` to specify from which incident the attribute value should be compared. For example, you might specify `${parent.hostedOn}` or `${child.ifDescr}`. Possible Source Object choices are as follows: • Card [click here for a list of attribute values] **Unique Keys from the Card Form: Capabilities Tab**: ▪ capability (Unique Key of the Capability) • Interface [click here for a list of attribute values] Use the following syntax to specify a Custom Attribute (CA) for an Interface: `valueOfInterfaceCa(<`*`CA_Name`*`>)` For example: `${child.valueOfInterfaceCA(Role)} = WAN Connection` **Values from the Basics Attributes listed on the Interface Form**: ▪ hostedOn (Hosted On Node) You must use the full DNS name for the hostedOn value. **Values from the Interface Form: General Tab**: ▪ ifName (name configured for the interface) ▪ ifAlias (alias configured for the interface) ▪ ifDescr (description configured for the interface) ▪ ifIndex (index assigned to the interface) ▪ ifSpeed (speed configured for the interface) When entering the value for ifSpeed, use the actual numeric value for the interface speed. For example, use `10000000` for ifSpeed **10 Mbps**. **Addresses from the Interface Form: IP Addresses Tab**: ▪ ipAddress (IP Address associated with the interface) Because NNMi uses a lexicographical compare when evaluating IP addresses, it is recommended that you use the **like** and **not like** operators to specify IP

Valid Attributes , continued

Attribute	Description
	address ranges rather than using the following operators: **>**, **>=**, **<**, or **<=**. **Unique Keys from the Interface Form: Capabilities Tab**: ▪ capability (Unique Key of the Capability) **Values from the Basics Attributes on the parent Node Form**: ▪ isSnmpInterface (Agent Enabled) **Values from the parent Node Form: General Tab**: ▪ sysOidInterface (System Object ID) **Values from the Basics Attributes on the associated Device Profile Form**: ▪ devVendorInterface (Device Vendor) ▪ devFamilyInterface (Device Family) • IP Address [click here for a list of attribute values] **Unique Keys from the IP Address Form: Capabilities Tab**: ▪ capability (Unique Key of the Capability) • Node [click here for a list of attribute values] Use the following syntax to specify a Custom Attribute (CA) for a Node: `valueOfNodeCa(<CA_Name>)` For example: `${valueOfNodeCa(Location)} = USA` **Values from the Basics Attributes on the Node Form**: ▪ hostname (Hostname, *case-sensitive*) ▪ mgmtIPAddress (Management Address) ▪ isSnmpNode (Agent Enabled) ▪ isNnmSystemLocal (NNMi Management Server) **Values from the Node Form: General Tab**: ▪ sysName (System Name) ▪ sysContact (System Contact) ▪ sysLocation (System Location) ▪ sysOidNode (System Object ID) **Addresses from the Node Form: IP Addresses Tab**: ▪ hostedIPAddress (Address) Because NNMi uses a lexicographical compare when evaluating IP addresses, it is recommended that you use the **like** and **not like** operators to specify IP

Valid Attributes , continued

Attribute	Description
	address ranges rather than using the following operators: **>**, **>=**, **<**, or **<=** . **Unique Keys from the Node Form: Capabilities Tab**: - capability (Unique Key of the Capability) **Values from the Basics Attributes on the associated Device Profile Form**: - devVendorNode (Device Vendor) - devFamillyNode (Device Family) **Values from the associated entry on the Regional Manager Form: Connection Tab**: - nnmSystemName (Hostname, *case-sensitive*) (*NNMi Advanced*) If the Global Network Management feature is enabled, this attribute value identifies a Regional Manager (NNMi management server). See "Global Manager: Connect to a Regional Manager" on page 103.

Valid Operator Values

Operator	Description

=	Finds all values equal to the value specified.
	Click here for examples.
	Match any incident with a CIA value of 5 and Object Identifier (OID) of .1.3.6.1.4.1.9.9.106.2.0.1:
	`${valueOfCia(\Q.1.3.6.1.4.1.9.9.106.2.0.1\E)} = 5`
	Match any incident with the Source Object's Capability equal to `com.hp.nnm.capability.card.fru`
	`$(capability) =  com.hp.nnm.capability.card.fru`

!=	Finds all values not equal to the value specified. Click here for an example. Match any incident with Device Vendor for the interface (Source Object) not equal to `Cisco`: `${devVendorInterface} != Cisco`
<	Finds all values less than the value specified. Click here for an example. Match any incident with a CIA value of less than 5 and Object Identifier (OID) of .1.3.6.1.4.1.9.9.106.2.0.1:

Valid Operator Values, continued

Operator	Description
	`${valueOfCia(\Q.1.3.6.1.4.1.9.9.106.2.0.1\E)} < 5`
<=	Finds all values less than or equal to the value specified. Click here for examples. Match any incident with a CIA value of less than or equal to 5 and Object Identifier (OID) of .1.3.6.1.4.1.9.9.106.2.0.1: `${valueOfCia (\Q.1.3.6.1.4.1.9.9.106.2.0.1\E)} <= 5`
>	Finds all values greater than the value specified. Click here for an example. Match any incident with a CIA value of greater than 5 and Object Identifier (OID) of .1.3.6.1.4.1.9.9.106.2.0.1: ${valueOfCia (\Q.1.3.6.1.4.1.9.9.106.2.0.1\E)} > 5
>=	Finds all values greater than or equal to the value specified. Click here for an example. Match any incident with a Source Object's (interface speed) ifSpeed value of 10Mbps: `${ifSpeed} >= 10000000`
is not null	Finds all non-blank values. Click here for an example. Match any incident with a Source Object's (interface name) ifName attribute that contains a value: `${ifName} is not null`
is null	Finds all blank values. Click here for an example. Match any incident with a Source Object's (interface name) ifName attribute that does not contain a value: `${ifName} is null`
like	Finds matches using the syntax defined for Java regular expressions. For more information, see the Pattern (Java Platform SE6) API documentation at: `http:/http://download.oracle.com/javase/6/docs/api/java/util/regex/Pattern.html` **Note:** To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E . The period asterisk (.*) characters mean *any number of characters of any type at this*

Valid Operator Values, continued

Operator	Description

	location.
	The period (.) character means *any single character of any type at this location*.
	Click here for an example.
	Match any incident with a Source Object's (interface description) ifDescr attribute that includes `Serial` followed by one or more digits:
	`${ifDescr} like Serial\d+`
	Match any incident with a Source Object's (interface alias) ifAlias attribute that contains `EtherChannel` (for example, `PAgPEtherChannel Group 1`).
	Note: The . (period) indicates any alphanumeric character.
	`${ifAlias} like .*EtherChannel.*`
	Match any incident with a CIA attribute value of `Chassis Fan Tray` followed by a digit and Object Identifier (OID) of `.1.3.6.1.4.1.9.9.13.1.4.1.3`
	Note: To include literal strings in the value, enclose the string value in `\Q<literal_value>\E` as shown in the following example.
	`${valueOfCia(\Q.1.3.6.1.4.1.9.9.13.1.4.1.3\E)} like Chassis Fan Tray \d`

not like	Finds all matches that do not have the values specified using the syntax defined for Java regular expressions. For more information,see the Pattern (Java Platform SE6) API documentation at : `http://download.oracle.com/javase/6/docs/api/java/util/regex/Pattern.html` **Note:** To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E . The period asterisk (.*) characters mean *any number of characters of any type at this location*. The period (.) character means *any single character of any type at this location*. Click here for an example. Match any incident with a Source Object's (interface name) ifName value that does not include `rtr`: `${ifName} not like .*rtr.*`

Valid Expressions

Attribute	Description
Expression	The value or pattern for which you want NNMi to search. Note the following: • The expression can include a valid Attribute. • The value or pattern you want to match is case sensitive. • When entering the value for ifSpeed, use the actual numeric value for the interface speed. For example, use `10000000` for ifSpeed **10 Mbps**.

Configure a Correlation Filter

Note: See **Help** → **Documentation Library** → **Release Notes**, and locate the **Support Matrix** link for Correlation Filter limitations.

When correlating groups of incidents under a Parent incident, you must specify the Correlation Filter that defines the relationship requirements that must be met before the incidents are correlated. The Correlation Filter tab enables you to use the Filter Editor to define these relationship requirements. See Valid Operators in the table that follows for examples of valid Correlation Filters.

Use the Filter Editor Buttons to insert Boolean Operators and to append, insert, and replace expressions in the Filter String. Use the Drag and Drop feature to make changes to the placement of the expressions in your Filter String. Click here for more information about using the Filter Editor for Custom Correlations:

- You can use Custom Incident Attributes, attributes for an incident's Source Node or Source Object, or both to define how matching incidents should be considered for the Correlation Rule. See Valid Attributes for more information.
- When specifying Attribute names and values, NNMi uses the type to determine a match. For example, if the Attribute type is numeric, NNMi does a numeric comparison. If the Attribute type is textual, NNMi does a lexographical string comparison. In all cases, when you use the **like** or **not like** operator, NNMi uses a lexographical string comparison. Click here for more information about Attribute types:
  - `ifIndex` and `ifSpeed` are numeric Attributes.
  - Any Attribute name that begins wtih "is" (`isSnmpInterface`, `isSnmpNode`, `isNnmSystemLocal`) represents a Boolean Attribute.
  - All other Attributes are textual.
- Each set of expressions associated with a Boolean Operator (for example, `AND`) is treated as if it were enclosed in parentheses and evaluated together.
- View the expression displayed under **Filter String** to see the logic of the expression as it is created.
- The `AND` and `OR` Boolean Operators must contain at least two expressions.

- The placement of your cursor and the subsequent text that is selected is important when performing operations using the Additional Filters Editor. For example, you append to or replace, the expression that is selected.
- The placement of your cursor and the subsequent text that is selected is especially important when adding your Boolean operators. See "Add Boolean Operators in the Additional Filters Editor" on page 310 for more information.

Filter Editor Buttons and Drag and Drop Feature

Button or Feature	Description
Append	Appends the current expression (Attribute, Operator,and Value) to the selected expression already included in the filter string.
Insert	Inserts the current expression (Attribute, Operator,and Value) in front of the cursor location within the Filter String.
Replace	Replaces the selected expression with the expression displayed Left or Right Expression.
AND	Inserts the AND Boolean Operator in the selected cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
OR	Inserts the OR Boolean Operator in the current cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
Delete	Deletes the selected expression. **Note**: If the Boolean Operator is selected, the Filter Editor deletes all expressions associated with the Boolean Operator.
Drag and Drop	You can drag any of the following items to a new location in the Filter String: ▪ Filter Editor Options: AND, OR, NOT, EXISTS, NOT EXISTS ▪ Filter Expression (Attribute, Operator and Value) When moving items in the Filter String, note the following: ▪ Click the item you want to move before dragging it to a new location. ▪ As you drag a selected item, an underline indicates the target location. ▪ If you are moving the selection up, NNMi places the item above the target location. ▪ If you are moving the selection down, NNMi places the item below the target location. ▪ If you attempt to move the selection to an invalid target location, NNMi displays an error message.

See "Correlation Rule Example" on page 710 for a step-by-step example of how the Subinterface Custom Correlation Rule provided by NNMi was created.

To configure a Correlation Filter:

1. Navigate to the **Custom Correlation Configuration** form:
   a. From the workspace navigation pane, select the **Configuration** workspace.
   b. Expand the **Incidents** folder.
   c. Select **Custom Correlation Configuration**.
2. Navigate to the **Correlation Rules** tab.
3. From the **Correlation Rules** table toolbar, do one of the following:
   - To create a Correlation Rule, click the ✳ New icon, and continue.
   - To edit a Correlation Rule, click the Open icon in the row representing the Correlation Rule you want to edit, and continue.
   - To delete a Correlation Rule, click the ✖ Delete icon.
4. Navigate to the **Correlation Filter** tab.
5. Create your Correlation Filter (see Filter Editor Components).

   Filter Editor Components

Component	Description
Attribute	The Attribute on which NNMi searches. See Valid Attributes below for a description of valid Attributes.
Operator	Use this Operator to establish the relationship between the Attribute and Expression. See Valid Operators in the table below for the description of each valid Operator.
Expression	Use the Expression to complete the criteria for the required relationship between the parent and child incident configurations. See Valid Expressions below for a description of the components that you might include in your Right Expression.

6. Click **Save and Close** to save your changes and return to the previous form.

Valid Attributes

Attribute	Description
Attribute	The Attribute on which NNMi searches. Valid attributes other than Source Node attributes depend on the Incident's Source Object. NNMi checks the Source Node as well as the Source Object for any Capability value. Note the following when specifying Attributes: • Boolean Attributes begin with "is" and must contain the value `true` or `false`.

Valid Attributes , continued

Attribute	Description
	• Use the following syntax to specify a Custom Incident Attribute (CIA): `valueOfCia(<CIA_Name>)` ▪ Check the appropriate Incident form for any valid CIA Names provided by NNMi. For example: `${valueOfCia(\Q.1.3.6.1.4.1.9.9.106.2.0.1\E)}= 5` ▪ When specifying the <*CIA_Name*>, you can use the syntax defined for Java regular expressions. For example, use the following syntax to specify that you want to include any CIA Name that begins with .1.3.6.1.2.1.31.1.1.1.1.: `${valueOfCia(\Q.1.3.6.1.2.1.31.1.1.1.1.\E.*)}` ▪ Enclose all CIA names using the `\Q` and `\E` characters so that NNMi correctly interprets the period character. For example: `${child.valueOfCia(\Qcia.address\E)}` For more information, see the Pattern (Java Platform SE6) API documentation at `http://download.oracle.com/javase/6/docs/api/java/util/regex/Pattern.html` • If you use attributes that are valid for the Source Node, NNMi uses the Source Node when comparing values. If you use attributes that are valid for the Source Object, NNMi uses the Source Object when comparing values. You cannot use attributes that are valid for the Source Node and Source Object in the same filter. • When using attributes for a Source Object, note the following: ▪ When using attributes for a Source Object, the attribute must be valid for the incident's Source Object or NNMi does not find a match. For example, if you use the `hostedOn` attribute and the Source Object is not an interface, the correlation does not occur. **Tip:** To check a Source Object for an incident, select the incident of interest, then select Open from the Lookup menu for the Source Object, and examine the Source Object form. ▪ *SNMP Trap incidents only*. NNMi does not find a match when the value for a Source Object is `None`. A Source Object attribute value of `None` indicates that NNMi cannot resolve the Source Object. If you want to match the incident, use one or more Source Node attributes. • If the incident does not have a Source Object, a Source Node, or both (for example, the node is not stored in the NNMi database), you must use CIAs in your filter rather than Source Object or Source Node attributes. **Tip:** To check whether the Source Object or Source Node is stored in the NNMi database, open the incident and then select Open from the Lookup menu for the Source Node or Source Object displayed. If a form does not open for the

Valid Attributes , continued

Attribute	Description
	selected object or node, this means the source is not stored in the NNMi database. • When specifying a Correlation Filter, precede the attribute name with either `parent.` or `child.` to specify from which incident the attribute value should be compared. For example, you might specify `${parent.hostedOn}` or `${child.ifDescr}`. Possible Source Object choices are as follows: • Card [click here for a list of attribute values] **Unique Keys from the Card Form: Capabilities Tab**: ▪ capability (Unique Key of the Capability) • Interface [click here for a list of attribute values] Use the following syntax to specify a Custom Attribute (CA) for an Interface: `valueOfInterfaceCa(<CA_Name>)` For example: `${child.valueOfInterfaceCA(Role)} = WAN Connection` **Values from the Basics Attributes listed on the Interface Form**: ▪ hostedOn (Hosted On Node) You must use the full DNS name for the hostedOn value. **Values from the Interface Form: General Tab**: ▪ ifName (name configured for the interface) ▪ ifAlias (alias configured for the interface) ▪ ifDescr (description configured for the interface) ▪ ifIndex (index assigned to the interface) ▪ ifSpeed (speed configured for the interface) When entering the value for ifSpeed, use the actual numeric value for the interface speed. For example, use `10000000` for ifSpeed **10 Mbps**. **Addresses from the Interface Form: IP Addresses Tab**: ▪ ipAddress (IP Address associated with the interface) Because NNMi uses a lexicographical compare when evaluating IP addresses, it is recommended that you use the **like** and **not like** operators to specify IP address ranges rather than using the following operators: **>**, **>=**, **<**, or **<=**. **Unique Keys from the Interface Form: Capabilities Tab**:

Valid Attributes , continued

Attribute	Description
	■ capability (Unique Key of the Capability) **Values from the Basics Attributes on the parent Node Form**: ■ isSnmpInterface (Agent Enabled) **Values from the parent Node Form: General Tab**: ■ sysOidInterface (System Object ID) **Values from the Basics Attributes on the associated Device Profile Form**: ■ devVendorInterface (Device Vendor) ■ devFamilyInterface (Device Family) • IP Address [click here for a list of attribute values] **Unique Keys from the IP Address Form: Capabilities Tab**: ■ capability (Unique Key of the Capability) • Node [click here for a list of attribute values] Use the following syntax to specify a Custom Attribute (CA) for a Node: `valueOfNodeCa(<CA_Name>)` For example: `${valueOfNodeCa(Location)} = USA` **Values from the Basics Attributes on the Node Form**: ■ hostname (Hostname, *case-sensitive*) ■ mgmtIPAddress (Management Address) ■ isSnmpNode (Agent Enabled) ■ isNnmSystemLocal (NNMi Management Server) **Values from the Node Form: General Tab**: ■ sysName (System Name) ■ sysContact (System Contact) ■ sysLocation (System Location) ■ sysOidNode (System Object ID) **Addresses from the Node Form: IP Addresses Tab**: ■ hostedIPAddress (Address) Because NNMi uses a lexicographical compare when evaluating IP addresses, it is recommended that you use the **like** and **not like** operators to specify IP address ranges rather than using the following operators: **>**, **>=**, **<**, or **<=** . **Unique Keys from the Node Form: Capabilities Tab**:

Valid Attributes , continued

Attribute	Description
	- capability (Unique Key of the Capability) **Values from the Basics Attributes on the associated Device Profile Form**: - devVendorNode (Device Vendor) - devFamillyNode (Device Family) **Values from the associated entry on the Regional Manager Form: Connection Tab**: - nnmSystemName (Hostname, *case-sensitive*) (*NNMi Advanced*) If the Global Network Management feature is enabled, this attribute value identifies a Regional Manager (NNMi management server). See "Global Manager: Connect to a Regional Manager" on page 103.

Valid Operators

Operator	Description
=	Finds all values equal to the value specified. Click here for an example. Correlate the incidents if the `hostedOn` value for the Source Object of the Child Incident is equal to the `hostedOn` value for the Source Object in the Parent Incident. `${child.hostedOn} = ${parent.hostedOn}`
!=	Finds all values not equal to the value specified. Click here for an example. Correlate the incidents if the `hostedOn` value for the Source Object of the Child Incident is not equal to the `hostedOn` value for the Source Object in the Parent Incident. `${child.hostedOn} != ${parent.hostedOn}`
<	Finds all values less than the value specified.
<=	Finds all values less than or equal to the value specified.
>	Finds all values greater than the value specified.
>=	Finds all values greater than or equal to the value specified.
is not null	Finds all non-blank values.
is null	Finds all blank values.
like	Finds matches using the syntax defined for Java regular expressions. See the

Valid Operators, continued

Operator	Description
	Pattern (Java Platform SE6) API documentation at : http://download.oracle.com/javase/6/docs/api/java/util/regex/Pattern.html for more information. **Note**: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E . The period asterisk (.*) characters mean *any number of characters of any type at this location*. The period (.) character means *any single character of any type at this location*.
not like	Finds all matches that do not have the values specified using the syntax defined for Java regular expressions. See the Pattern (Java Platform SE6) API documentation at : http://download.oracle.com/javase/6/docs/api/java/util/regex/Pattern.html for more information. **Note**: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E . The period asterisk (.*) characters mean *any number of characters of any type at this location*. The period (.) character means *any single character of any type at this location*.

Valid Expressions

Attribute	Description
Expression	The value or pattern for which you want NNMi to search. Note the following: • The expression can include a valid Attribute. • The value or pattern you want to match is case sensitive. • When entering the value for ifSpeed, use the actual numeric value for the interface speed. For example, use `10000000` for ifSpeed **10 Mbps**.

Correlation Rule Example

Tip: Use these steps as a guideline for creating your own Correlation Rules.

This example uses the Subinterface Correlation Rule to describe the steps for creating a Correlation Rule. The Subinterface Correlation Rule specifies that Interface Down incidents that occur for subinterfaces should be correlated under the Interface Down incident generated for the main interface. Click here for more information about Custom Correlations.

The NNMi Custom Correlation feature enables you to correlate groups of incidents under a Parent Incident. This feature is useful when you want to define a relationship between a number of incidents potentially from different network objects that form a logical set to identify a problem. The set of correlations is considered complete if all of the incidents arrive within a specified time window. You can correlate incidents under an existing Incident Configuration (Correlation Rule) or create a new Incident Configuration (Causal Rule).

This example uses an existing Incident Configuration as the Parent Incident. See "Causal Rule Example" on page 744 for an example that generates a new Incident Configuration as the Parent Incident.

To configure the Subinterface Correlation Rule Basics information:

1. Navigate to the **Custom Correlation Configuration** form:
   a. From the workspace navigation pane, select the **Configuration** workspace.
   b. Expand the **Incidents** folder.
   c. Select **Custom Correlation Configuration**.
2. Navigate to the **Correlation Rules** tab.
3. From the **Correlation Rules** table toolbar, click the * New icon.
4. In the **Name** attribute, enter a unique name that will help you to identify the Correlation Rule. In this example, the Correlation Rule Name is **Subinterface**.
5. In the **Author** attribute, enter a name that identifies the person who is creating the Correlation Rule. In this example, **HP Network Node Manager** is the Author name to identify this Correlation Rule as one that NNMi provides.
6. Make sure **Enabled** ☑ is checked to indicate the NNMi Causal Engine should use this Correlation Rule when evaluating incidents.
7. To use an existing Parent Incident, do the following:
   a. In the **Parent Incident** Lookup Field, select Quick Find to select from the list of existing incident configurations.
   b. In the Subinterface Correlation Rule, the **InterfaceDown** incident configuration was selected as the Parent Incident.
8. Select the Incident Configuration that must match an incoming incident and that should be correlated as the Child Incident for the Custom Correlation.

   In the Subinterface Correlation Rule, the **InterfaceDown** incident configuration was also selected as the Child Incident.
9. In the **Correlation Window Duration** attribute, enter the time limit (in days, hours, minutes, and seconds) that must be reached before the incoming incident are correlated. The Subinterface Correlation Rule specifies a Correlation Window Duration of 6 minutes.
10. Use the **Description** attribute to provide additional information that you would like to store about the current incident configuration. This description applies only to the configuration entry.

    The Subinterface Correlation Rule includes the following description: **Correlates sub-interfaces down incidents under the main interface down**.

To configure the Parent Incident Filter:

1. In the Correlation Rule form, navigate to the **Parent Incident Filter** tab.
2. The following Parent Incident Filter specifies that the Correlation Rule applies only to Cisco

devices:

```
${devVendorInterface} = Cisco
```

3. The following Parent Incident Filter specifies that the ifDescr value must contain the string `Serial` followed by one or more digits and then a forward slash, followed by zero or more digits:

   ```
   ${ifDescr}like Serial\d+/*\d*
   ```

4. As shown in the following Filter String, the Parent Incident Filters use the Boolean operator AND so that both criteria must be met for the Incident to be selected as a Parent:

   ```
   {${devVendorInterface} = Cisco AND ${ifDescr} like Serial\d+/*\d*)
   ```

   To create this Parent Incident Filter, in the Filter Editor:

   a. Click **And**.

   b. In the **Attribute** field, enter `${devVendorInterface}`.

   c. In the **Operator** field, select **=** from the drop-down menu.

   d. In the **Expression** field, enter **Cisco**.

   e. Click **Append**.

   f. In the **Attribute** field, enter `${ifDescr}`.

   g. In the **Operator** field, select `like` from the drop-down menu.

   h. In the **Expression** field, enter `Serial\d+/*\d*`.

   i. Click **Add**.

To configure the Child Incident Filter:

1. In the Correlation Rule form, navigate to the **Child Incident Filter** tab.

2. The following Child Incident Filter specifies that the Correlation Rule applies only to Cisco devices:

   ```
   ${devVendorInterface} = Cisco
   ```

3. The following Child Incident Filter specifies that the ifDescr value must contain the following sequence of values:

   The string `Serial` followed by one or more digits, then a forward slash, followed by zero or more digits, and then a period followed by one or more digits:

   ```
   ${ifDescr}like Serial\d+/*\d*
   ```

4. As shown in the following Filter String, the Child Incident Filters use the Boolean operator AND so that both criteria must be met for the Incident to be selected as a Child:

   ```
   {${devVendorInterface} = Cisco AND ${ifDescr} like Serial\d+/*\d*)
   ```

   To create this Parent Incident Filter, in the Filter Editor:

   a. Click **And**.

   b. In the **Attribute** field, enter `${devVendorInterface}`.

c. In the **Operator** field, select **=** from the drop-down menu.

d. In the **Expression** field, enter **Cisco**.

e. Click **Append**.

f. In the **Attribute** field, enter `${ifDescr}`.

g. In the **Operator** field, select `like` from the drop-down menu.

h. In the **Expression** field, enter `Serial\d+/*\d*`.

i. Click **Append**.

To configure the Correlation Filter:

Note: When specifying a Correlation Filter, you must specify whether the attribute is from a Child Incident or Parent Incident using the following syntax: `${child.<`*`attribute_name`*`>}` or `${parent.<`*`attribute_name`*`>}`.

1. In the Correlation Rule form, navigate to the **Correlation Filter** tab.

2. To ensure that the Interface Down incidents are generated for the same node, the Subinterface Correlation Rules uses `hostedOn` as the attribute for both the Child and Parent Incidents as shown in the following example filter:

   `${child.hostedOn}= ${parent.hostedOn}`

   To ensure that the Interfaces are subinterfaces for the main interface, the filter also matches the ifDescr values:

   `${child.ifDescr}like ${parent.ifDescr}.*`

   As shown in the following Filter String, the Correlation Filter uses the Boolean operator AND so that both criteria must be met for the Incidents to be correlated:

   `${child.hostedOn} = ${parent.hostedOn}`**AND** `${child.ifDescr} like ${parent.ifDescr}.*`

   To create the Correlation Rule filter:

   a. Click **And**.

   b. In the **Attribute** field, enter `${child.hostedOn}`.

   c. In the **Operator** field, select **=** from the drop-down menu.

   d. In the **Expression** field, enter ${parent.hostedOn}.

   e. Click **Append**.

   f. In the **Attribute** field, enter `${child.ifDescr}`.

   g. In the **Operator** field, select `like` from the drop-down menu.

   h. In the **Expression** field, enter `${parent.ifDescr}.*`.

   i. Click **Append**.

3. Click **Save and Close** to save your changes and return to the previous form.

Configure a Causal Rule

Tip: Configure a Causal Rule when you want to cause NNMi to generate a Parent Incident and you want to correlate one or more Child Incident Configurations under the Parent Incident that you cause to be generated.

Note: See **Help** → **Documentation Library** → **Release Notes**, and locate the **Support Matrix** link for Causal Rule limitations.

When correlating groups of incidents under a Parent incident, use the Causal Rules tab to specify the following.

- Parent Incident Configuration to be generated
- One or more Child Incident Configurations to be correlated with the generated Parent Incident
- Filters that NNMi should use when selecting the Child Incident instances for correlation
- Source Object and Source Node Filter to be used to determine the Source Node and Source Object for the Parent Incident that is generated
- The time window that must be met before NNMi correlates the incidents

For information about each Causal Rules tab:

To configure a Causal Rule:

1. Navigate to the **Custom Correlation Configuration** form:
   a. From the workspace navigation pane, select the **Configuration** workspace.
   b. Expand the **Incidents** folder.
   c. Select **Custom Correlation Configuration**.
2. Navigate to the **Causal Rules** tab.
3. From the **Causal Rules** table toolbar, do one of the following:
   - To create a Causal Rule, click the New icon, and continue.
   - To edit a Causal Rule, click the Open icon in the row representing the Causal Rule you want to edit, and continue.
   - To delete a Causal Rule, click the Delete icon.
4. Create your Causal Rule (see table).
5. Click **Save and Close** to save your changes and return to the previous form.

Cause Rule Basic Attributes

Attribute	Description
Name	Type a maximum of 64 characters. Alpha-numeric, spaces, and special characters (~ ! @ # $ % ^ &amp; * ( ) _+ -) are permitted. The name is used to identify the Causal Rule and must be unique. Use a name that

Cause Rule Basic Attributes, continued

Attribute	Description
	will help you to remember the purpose of the Causal Rule.
Author	Indicates who created or last modified the Causal Rule. **Caution:** If the Author attribute value is **HP Network Node Manager**, any changes are at risk of being overwritten in the future. • Click **Lookup** and select **Show Analysis** to display details about the currently selected Author. • Click **Quick Find** to access the list of existing Author values. • Click **New** to create an Author value.
Enabled	If ☑ enabled, the NNMi Causal Engine uses the Causal Rule when evaluating incidents. If □ disabled, the Causal Rule is ignored.
Parent Incident	Specifies the incident configuration that should be generated as the Parent Incident for the Causal Rule. **To specify a Parent Incident configuration**: 1. Click the Lookup icon, and do one of the following: ▪ To display Analysis Pane information, select Show Analysis. (See Use the Analysis Pane for more information about the Analysis Pane.) ▪ To display the list of possible incidents, select Quick Find. In the Quick Find dialog, select the Incident of interest. ▪ To create a Parent Incident, select one of the following: ○ New Management Event Configuration ○ New Remote NNM Event Configuration ○ New SNMP Trap Configuration ▪ To modify a Parent Incident, select Open. 2. *Optional*. To create or modify a Parent Incident, enter or modify the attribute values for the selected Incident configuration. See "Configuring Incidents" on page 589 for more information about the Incident Configuration form. 3. Click **Save and Close** to save your changes and return to the previous form. 4. *Optional*. To create or modify a Parent Incident, enter or modify the attribute values for the selected Incident configuration. See "Configuring Incidents" on page 589 for more information about the Incident Configuration form. 5. Click **Save and Close** to save your changes and return to the previous form.

Cause Rule Basic Attributes, continued

Attribute	Description
Correlation Nature	Select the Correlation Nature that you want to assign to the Parent Incident that is generated. **Note**: The Child Incident will have the Correlation Nature of Secondary Root Cause.
Common Child Incident Attribute	Specifies the Incident Attribute that all Child Incidents must have in common for the incident instance to be correlated under the Parent Incident defined for the Causal Rule. For example, if you want to ensure that all child incidents are from the same node, use the `${hostedOn}` attribute. Valid attributes other than Source Node attributes depend on the Incident's Source Object. NNMi checks the Source Node as well as the Source Object for any attribute value. Note the following when specifying Attributes: • You cannot specify $(capability) as a Common Child Incident Attribute. • Boolean Attributes begin with "is" and must contain the value `true` or `false`. • Use the following syntax to specify a Custom Incident Attribute (CIA): `valueOfCia(<`*`CIA_Name`*`>)` **Note**: Check the appropriate Incident form for any valid CIA Names provided by NNMi. For example: `${valueOfCia(\Q.1.3.6.1.4.1.9.9.106.2.0.1\E)}` • When specifying the <*CIA_Name*>, you can use the syntax defined for Java regular expressions. For example, use the following syntax to specify that you want to include any CIA Name that begins with .1.3.6.1.2.1.31.1.1.1.1.: `${valueOfCia(\Q.1.3.6.1.2.1.31.1.1.1.1.\E.*)}` **Note**: Enclose all CIA names using the `\Q` and `\E` characters so that NNMi correctly interprets the period character. For example: `$child.valueOfCia(\Qcia.address\E)}`. See the Pattern (Java Platform SE6) API documentation at: `http://download.oracle.com/javase/6/docs/api/java/util/regex/Pattern.html` for more information. • If you use attributes that are valid for the Source Node, NNMi uses the Source Node when comparing values. If you use attributes that are valid for the Source Object, NNMi uses the Source Object when comparing values. You cannot use attributes that are valid for the Source Node and Source Object in the same filter. • When using attributes for a Source Object, the attribute must be valid for the incident's Source Object or NNMi does not find a match. For example, if you use the `hostedOn` attribute and the Source Object is not an interface, the correlation does not occur. **Tip**: To check a Source Object for an incident, select the incident of interest, then

Cause Rule Basic Attributes, continued

Attribute	Description
	select Open from the Lookup menu for the Source Object, and examine the Source Object form. A Source Object attribute value of `None` indicates that NNMi cannot identify the Source Object or the Source Object is a Node. If you want to match the incident, use one or more Source Node attributes. • If the incident does not have a Source Object, a Source Node, or both (for example, the node is not stored in the NNMi database), you must use CIAs rather than Source Object or Source Node attributes. **Tip**: To check whether the Source Object or Source Node is stored in the NNMi database, open the incident and then select Open from the Lookup menu for the Source Node or Source Object displayed. If a form does not open for the selected object or node, this means the source is not stored in the NNMi database. Possible Source Object choices are as follows: • Interface [click here for a list of attribute values] Use the following syntax to specify a Custom Attribute (CA) for an Interface: `valueOfInterfaceCa(`<*CA_Name*>`)` For example: `${child.valueOfInterfaceCA(Role)}` **Values from the Basics Attributes listed on the Interface Form**: ▪ hostedOn (Hosted On Node) **Values from the Interface Form: General Tab**: ▪ ifName (name configured for the interface) ▪ ifAlias (alias configured for the interface) ▪ ifDescr (description configured for the interface) ▪ ifIndex (index assigned to the interface) ▪ ifSpeed (speed configured for the interface) **Addresses from the Interface Form: IP Addresses Tab**: ▪ ipAddress (IP Address associated with the interface) **Values from the Basics Attributes on the parent Node Form**: ▪ isSnmpInterface (Agent Enabled) **Values from the parent Node Form: General Tab**: ▪ sysOidInterface (System Object ID) **Values from the Basics Attributes on the associated Device Profile Form**:

Cause Rule Basic Attributes, continued

Attribute	Description
	▪ devVendorInterface (Device Vendor) ▪ devFamilyInterface (Device Family) • Node [click here for a list of attribute values] Use the following syntax to specify a Custom Attribute (CA) for a Node: `valueOfNodeCa(<CA_Name>)` For example: `${valueOfNodeCa(Location)}` **Values from the Basics Attributes on the Node Form**: ▪ hostname (Hostname, *case-sensitive*) ▪ mgmtIPAddress (Management Address) ▪ isSnmpNode (Agent Enabled) ▪ isNnmSystemLocal (NNMi Management Server) **Values from the Node Form: General Tab**: ▪ sysName (System Name) ▪ sysContact (System Contact) ▪ sysLocation (System Location) ▪ sysOidNode (System Object ID) **Addresses from the Node Form: IP Addresses Tab**: ▪ hostedIPAddress (Address) **Values from the Basics Attributes on the associated Device Profile Form**: ▪ devVendorNode (Device Vendor) ▪ devFamillyNode (Device Family) **Values from the associated entry on the Regional Manager Form: Connection Tab**: ▪ nnmSystemName (Hostname, *case-sensitive*) (*NNMi Advanced*) If the Global Network Management feature is enabled, this attribute value identifies a Regional Manager (NNMi management server). See "Global Manager: Connect to a Regional Manager" on page 103.
Correlation Window Duration	The time window that must be met before NNMi correlates the incidents. Enter a number for Days, Hours, Minutes, and Seconds. Note the following: • NNMi waits until the Correlation Window Duration has passed before generating the Parent Incident and correlating its Child Incidents.

Cause Rule Basic Attributes, continued

Attribute	Description
	• If you are relating multiple Custom Correlations, make sure the Correlation Window Duration allows enough time for all of the Parent and Child incidents to be generated. For example, to correlate two or more Interface Down incidents under a new incident on interfaces that are polled every 5 minutes, use a 6-minute Correlation Window Duration. The 6-minute window ensures that the Interface Down incidents, which might occur 5 minutes apart, will be correlated under the new incident.
Description	Use the description field to provide additional information that you would like to store about the current incident configuration. This description applies only to the configuration entry. Type a maximum of 2048 characters. Alpha-numeric, spaces, and special characters (~ ! @ # $ % ^ & * ( ) _+ -) are permitted.

Configure a Child Incident for a Causal Rule

The Child Incident tab enables you to specify which Child Incidents should be considered for correlation according to the Causal Rule you are configuring.

For information about each Causal Rules tab:

For information about each Child Incident tab:

To configure a Child Incident for a Causal Rule:

1. Navigate to the **Custom Correlation Configuration** form:
   a. From the workspace navigation pane, select the **Configuration** workspace.
   b. Expand the **Incidents** folder.
   c. Select **Custom Correlation Configuration**.
2. Navigate to the **Causal Rules** tab.
3. From the **Causal Rules** table toolbar, do one of the following:
   - To create a Causal Rule, click the New icon, and continue.
   - To edit a Causal Rule, click the Open icon in the row representing the Causal Rule you want to edit, and continue.
   - To delete a Causal Rule, click the Delete icon.
4. Create your Causal Rule. (See "Configure a Causal Rule" on page 714.)
5. Create your Child Incident Configuration (see table).
6. *Optional*. Configure a Child Incident Filter. (See "Configure a Child Incident Filter for a Causal Rule" on page 721.)

7. *Optional*. Configure a Source Object Filter. (See "Configure a Source Object Filter for a Causal Rule" on page 730.)
8. *Optional*. Configure a Source Node Filter. (See "Configure a Source Node Filter for a Causal Rule" on page 738.)
9. Click **Save and Close** to save your changes and return to the previous form.

Causal Rule Basic Attributes

Attribute	Description
Name	Type a maximum of 64 characters. Alpha-numeric, spaces, and special characters (~ ! @ # $ % ^ &amp; * ( ) _+ -) are permitted. The name is used to identify the Child Incident Configuration and must be unique within each Causal Rule. Use a name that will help you to remember the purpose of the Child Incident Configuration.
Child Incident	Specifies the incident configuration that should be used as the Child Incident when evaluating the Causal Rule. **To specify a Child Incident configuration**: 1. Click the Lookup icon, and do one of the following: ▪ To specify a Child Incident without making any changes to the incident configuration, select Quick Find . In the Quick Find dialog, select the Incident of interest. ▪ To create a Child Incident, select one of the following: ○ New Management Event Configuration ○ New Remote NNM Event Configuration ○ New SNMP Trap Configuration ▪ To modify a Child Incident, select Open. 2. *Optional*. To create or modify a Child Incident, enter or modify the attribute values for the selected Incident configuration. See "Configuring Incidents" on page 589 for more information about the Incident Configuration form. 3. Click **Save and Close** to save your changes and return to the previous form.
Forward Child Custom Incident Attributes	Enter a comma-delimited list of the Custom Incident Attributes you want to appear with the generated Parent Incident. NNMi forwards these values from the Child Incidents that you configure for the Causal Rule.
Optional Child Incident	If ☑ enabled, the NNMi Causal Engine generates the Parent Incident whether this Child Incident occurs. If □ disabled, the NNMi Causal Engine only generates the Parent Incident if this Child Incident occurs.

Causal Rule Basic Attributes, continued

Attribute	Description
Use Child Incident's Source Object for Parent	If ☑ enabled, indicates you want NNMi to use the Source Object of the Child Incident as the Source Object for the Parent Incident. **Note**: If you enable this option, NNMi ignores any Source Object Filter you configured. If □ disabled, indicates you want NNMi to use the Source Object Filter configuration to determine the Parent Incident's Source Node. See "Configure a Source Object Filter for a Causal Rule" on page 730 for more information. If you do not specify the Source Object to use for the Parent Incident, NNMi uses the Source Object of the first Child Incident that occurs.
Use Child Incident's Source Node for Parent	If ☑ enabled, indicates you want NNMi to use the Source Node of the Child Incident as the Source Node for the Parent Incident. **Note**: If you enable this option, NNMi ignores any Source Node Filter you configured. If □ disabled, indicates you want NNMi to use the Source Node Filter configuration to determine the Parent Incident's Source Node. See "Configure a Source Node Filter for a Causal Rule" on page 738 for more information. If you do not specify the Source Node to use for the Parent Incident, NNMi uses the Source Node of the first Child Incident that occurs.

Configure a Child Incident Filter for a Causal Rule

Note: See **Help** → **Documentation Library** → **Release Notes**, and locate the **Support Matrix** link for Child Incident Filter limitations.

The Child Incident Filter tab enables you to create a filter to specify which Child Incidents should be considered for correlation according to the Causal Rule you are configuring.

For information about each Causal Rules tab:

For information about each Child Incident tab:

Use the Filter Editor Buttons to insert Boolean Operators and to append, insert, and replace expressions in the Filter String. Use the Drag and Drop feature to make changes to the placement of the expressions in your Filter String. Click here for more information about using the Filter Editor for Custom Correlations:

- You can use Custom Incident Attributes, attributes for an incident's Source Node or Source Object, or both to define how matching incidents should be considered for the Correlation Rule. See Valid Attributes for more information.
- When specifying Attribute names and values, NNMi uses the type to determine a match. For example, if the Attribute type is numeric, NNMi does a numeric comparison. If the Attribute type is textual, NNMi does a lexographical string comparison. In all cases, when you use the **like** or **not like** operator, NNMi uses a lexographical string comparison. Click here for more information

about Attribute types:

- ifIndex and ifSpeed are numeric Attributes.
- Any Attribute name that begins wtih "is" (isSnmpInterface, isSnmpNode, isNnmSystemLocal) represents a Boolean Attribute.
- All other Attributes are textual.

- Each set of expressions associated with a Boolean Operator (for example, AND) is treated as if it were enclosed in parentheses and evaluated together.
- View the expression displayed under **Filter String** to see the logic of the expression as it is created.
- The AND and OR Boolean Operators must contain at least two expressions.
- The placement of your cursor and the subsequent text that is selected is important when performing operations using the Additional Filters Editor. For example, you append to or replace, the expression that is selected.
- The placement of your cursor and the subsequent text that is selected is especially important when adding your Boolean operators. See "Add Boolean Operators in the Additional Filters Editor" on page 310 for more information.

Filter Editor Buttons and Drag and Drop Feature

Button or Feature	Description
Append	Appends the current expression (Attribute, Operator,and Value) to the selected expression already included in the filter string.
Insert	Inserts the current expression (Attribute, Operator,and Value) in front of the cursor location within the Filter String.
Replace	Replaces the selected expression with the expression displayed Left or Right Expression.
AND	Inserts the AND Boolean Operator in the selected cursor location. **Note:** View the expression displayed under **Filter String** to see the logic of the expression as it is created.
OR	Inserts the OR Boolean Operator in the current cursor location. **Note:** View the expression displayed under **Filter String** to see the logic of the expression as it is created.
Delete	Deletes the selected expression.

Filter Editor Buttons and Drag and Drop Feature, continued

Button or Feature	Description
	Note: If the Boolean Operator is selected, the Filter Editor deletes all expressions associated with the Boolean Operator.
Drag and Drop	You can drag any of the following items to a new location in the Filter String: ▪ Filter Editor Options: AND, OR, NOT, EXISTS, NOT EXISTS ▪ Filter Expression (Attribute, Operator and Value) When moving items in the Filter String, note the following: ▪ Click the item you want to move before dragging it to a new location. ▪ As you drag a selected item, an underline indicates the target location. ▪ If you are moving the selection up, NNMi places the item above the target location. ▪ If you are moving the selection down, NNMi places the item below the target location. ▪ If you attempt to move the selection to an invalid target location, NNMi displays an error message.

To configure a Child Incident Filter for a Causal Rule:

1. Navigate to the **Custom Correlation Configuration** form:

   a. From the workspace navigation pane, select the **Configuration** workspace.

   b. Expand the **Incidents** folder.

   c. Select **Custom Correlation Configuration**.

2. Navigate to the **Causal Rules** tab.

3. From the **Causal Rules** table toolbar, do one of the following:

   - To create a Causal Rule, click the ✱ New icon, and continue.

   - To edit a Causal Rule, click the Open icon in the row representing the Causal Rule you want to edit, and continue.

   - To delete a Causal Rule, click the ✖ Delete icon.

4. Create your Causal Rule. (See "Configure a Causal Rule" on page 714.)

5. Create your Child Incident Configuration . (See "Configure a Child Incident for a Causal Rule" on page 719.)

6. *Optional*. Configure a Child Incident Filter. (See Filter Editor Components).

Filter Editor Components

Component	Description
Attribute	The Attribute on which NNMi searches. See Valid Attributes below for a description of valid Attributes.
Operator	Use this Operator to establish the relationship between the Attribute and Expression. See Valid Operators in the table below for the description of each valid Operator.
Expression	Use the Expression to complete the criteria for the Parent Incident configuration. See Valid Expressions below for a description of the components that you might include in your Expression.

7. *Optional*. Configure a Source Object Filter. (See "Configure a Source Object Filter for a Causal Rule" on page 730.)
8. *Optional*. Configure a Source Node Filter. (See "Configure a Source Node Filter for a Causal Rule" on page 738.)
9. Click **Save and Close** to save your changes and return to the previous form.

Valid Attributes

Attribute	Description
Attribute	The Attribute on which NNMi searches. Valid attributes other than Source Node attributes depend on the Incident's Source Object. NNMi checks the Source Node as well as the Source Object for any Capability value. Note the following when specifying Attributes: • Boolean Attributes begin with "is" and must contain the value `true` or `false`. • Use the following syntax to specify a Custom Incident Attribute (CIA): `valueOfCia(<`*`CIA_Name`*`>)` ▪ Check the appropriate Incident form for any valid CIA Names provided by NNMi. For example: `${valueOfCia(\Q.1.3.6.1.4.1.9.9.106.2.0.1\E)}= 5` ▪ When specifying the <*CIA_Name*>, you can use the syntax defined for Java regular expressions. For example, use the following syntax to specify that you want to include any CIA Name that begins with .1.3.6.1.2.1.31.1.1.1.1.: `${valueOfCia(\Q.1.3.6.1.2.1.31.1.1.1.1.\E.*)}` ▪ Enclose all CIA names using the `\Q` and `\E` characters so that NNMi correctly interprets the period character. For example: `${child.valueOfCia(\Qcia.address\E)}` For more information, see the Pattern (Java Platform SE6) API documentation at `http://download.oracle.com/javase/6/docs/api/java/util/regex/Pattern.html`

Valid Attributes , continued

Attribute	Description
	• If you use attributes that are valid for the Source Node, NNMi uses the Source Node when comparing values. If you use attributes that are valid for the Source Object, NNMi uses the Source Object when comparing values. You cannot use attributes that are valid for the Source Node and Source Object in the same filter. • When using attributes for a Source Object, note the following: ▪ When using attributes for a Source Object, the attribute must be valid for the incident's Source Object or NNMi does not find a match. For example, if you use the `hostedOn` attribute and the Source Object is not an interface, the correlation does not occur. **Tip:** To check a Source Object for an incident, select the incident of interest, then select Open from the Lookup menu for the Source Object, and examine the Source Object form. ▪ *SNMP Trap incidents only*. NNMi does not find a match when the value for a Source Object is `None`. A Source Object attribute value of `None` indicates that NNMi cannot resolve the Source Object. If you want to match the incident, use one or more Source Node attributes. • If the incident does not have a Source Object, a Source Node, or both (for example, the node is not stored in the NNMi database), you must use CIAs in your filter rather than Source Object or Source Node attributes. **Tip:** To check whether the Source Object or Source Node is stored in the NNMi database, open the incident and then select Open from the Lookup menu for the Source Node or Source Object displayed. If a form does not open for the selected object or node, this means the source is not stored in the NNMi database. • When specifying a Correlation Filter, precede the attribute name with either `parent.` or `child.` to specify from which incident the attribute value should be compared. For example, you might specify `${parent.hostedOn}` or `${child.ifDescr}`. Possible Source Object choices are as follows: • Card [click here for a list of attribute values] **Unique Keys from the Card Form: Capabilities Tab**: ▪ capability (Unique Key of the Capability) • Interface [click here for a list of attribute values] Use the following syntax to specify a Custom Attribute (CA) for an Interface: `valueOfInterfaceCa(<`*`CA_Name`*`>)`

Valid Attributes , continued

Attribute	Description
	For example: `${child.valueOfInterfaceCA(Role)} = WAN Connection` **Values from the Basics Attributes listed on the Interface Form:** ▪ hostedOn (Hosted On Node) You must use the full DNS name for the hostedOn value. **Values from the Interface Form: General Tab:** ▪ ifName (name configured for the interface) ▪ ifAlias (alias configured for the interface) ▪ ifDescr (description configured for the interface) ▪ ifIndex (index assigned to the interface) ▪ ifSpeed (speed configured for the interface) When entering the value for ifSpeed, use the actual numeric value for the interface speed. For example, use `10000000` for ifSpeed **10 Mbps**. **Addresses from the Interface Form: IP Addresses Tab:** ▪ ipAddress (IP Address associated with the interface) Because NNMi uses a lexicographical compare when evaluating IP addresses, it is recommended that you use the **like** and **not like** operators to specify IP address ranges rather than using the following operators: **>**, **>=**, **<**, or **<=**. **Unique Keys from the Interface Form: Capabilities Tab:** ▪ capability (Unique Key of the Capability) **Values from the Basics Attributes on the parent Node Form:** ▪ isSnmpInterface (Agent Enabled) **Values from the parent Node Form: General Tab:** ▪ sysOidInterface (System Object ID) **Values from the Basics Attributes on the associated Device Profile Form:** ▪ devVendorInterface (Device Vendor) ▪ devFamilyInterface (Device Family) • IP Address [click here for a list of attribute values] **Unique Keys from the IP Address Form: Capabilities Tab:** ▪ capability (Unique Key of the Capability) • Node [click here for a list of attribute values] Use the following syntax to specify a Custom Attribute (CA) for a Node:

Valid Attributes , continued

Attribute	Description
	`valueOfNodeCa(<`*`CA_Name`*`>)` For example: `${valueOfNodeCa(Location)} = USA` **Values from the Basics Attributes on the Node Form**: ▪ hostname (Hostname, *case-sensitive*) ▪ mgmtIPAddress (Management Address) ▪ isSnmpNode (Agent Enabled) ▪ isNnmSystemLocal (NNMi Management Server) **Values from the Node Form: General Tab**: ▪ sysName (System Name) ▪ sysContact (System Contact) ▪ sysLocation (System Location) ▪ sysOidNode (System Object ID) **Addresses from the Node Form: IP Addresses Tab**: ▪ hostedIPAddress (Address) Because NNMi uses a lexicographical compare when evaluating IP addresses, it is recommended that you use the **like** and **not like** operators to specify IP address ranges rather than using the following operators: **>**, **>=**, **<**, or **<=** . **Unique Keys from the Node Form: Capabilities Tab**: ▪ capability (Unique Key of the Capability) **Values from the Basics Attributes on the associated Device Profile Form**: ▪ devVendorNode (Device Vendor) ▪ devFamillyNode (Device Family) **Values from the associated entry on the Regional Manager Form: Connection Tab**: ▪ nnmSystemName (Hostname, *case-sensitive*) (*NNMi Advanced*) If the Global Network Management feature is enabled, this attribute value identifies a Regional Manager (NNMi management server). See "Global Manager: Connect to a Regional Manager" on page 103.

Valid Operator Values

Operator	Description
=	Finds all values equal to the value specified.

Valid Operator Values, continued

Operator	Description

	Click here for examples.
	Match any incident with a CIA value of 5 and Object Identifier (OID) of .1.3.6.1.4.1.9.9.106.2.0.1:
	`${valueOfCia(\Q.1.3.6.1.4.1.9.9.106.2.0.1\E)} = 5`
	Match any incident with the Source Object's Capability equal to `com.hp.nnm.capability.card.fru`
	`$(capability) =  com.hp.nnm.capability.card.fru`

!=	Finds all values not equal to the value specified. Click here for an example. Match any incident with Device Vendor for the interface (Source Object) not equal to `Cisco`: `${devVendorInterface} != Cisco`
<	Finds all values less than the value specified. Click here for an example. Match any incident with a CIA value of less than 5 and Object Identifier (OID) of .1.3.6.1.4.1.9.9.106.2.0.1: `${valueOfCia(\Q.1.3.6.1.4.1.9.9.106.2.0.1\E)} < 5`
<=	Finds all values less than or equal to the value specified. Click here for examples. Match any incident with a CIA value of less than or equal to 5 and Object Identifier (OID) of .1.3.6.1.4.1.9.9.106.2.0.1: `${valueOfCia (\Q.1.3.6.1.4.1.9.9.106.2.0.1\E)} <= 5`
>	Finds all values greater than the value specified. Click here for an example. Match any incident with a CIA value of greater than 5 and Object Identifier (OID) of .1.3.6.1.4.1.9.9.106.2.0.1: ${valueOfCia (\Q.1.3.6.1.4.1.9.9.106.2.0.1\E)} > 5
>=	Finds all values greater than or equal to the value specified. Click here for an example. Match any incident with a Source Object's (interface speed) ifSpeed value of 10Mbps: `${ifSpeed} >= 10000000`
is not	Finds all non-blank values.

Valid Operator Values, continued

Operator	Description
null	Click here for an example. Match any incident with a Source Object's (interface name) ifName attribute that contains a value: `${ifName} is not null`
is null	Finds all blank values. Click here for an example. Match any incident with a Source Object's (interface name) ifName attribute that does not contain a value: `${ifName} is null`

like	Finds matches using the syntax defined for Java regular expressions. For more information, see the Pattern (Java Platform SE6) API documentation at: `http:/http://download.oracle.com/javase/6/docs/api/java/util/regex/Pattern.html`
	Note: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E .
	The period asterisk (.*) characters mean *any number of characters of any type at this location*.
	The period (.) character means *any single character of any type at this location*.
	Click here for an example.
	Match any incident with a Source Object's (interface description) ifDescr attribute that includes `Serial` followed by one or more digits:
	`${ifDescr} like Serial\d+`
	Match any incident with a Source Object's (interface alias) ifAlias attribute that contains `EtherChannel` (for example, `PAgPEtherChannel Group 1`).
	Note: The . (period) indicates any alphanumeric character.
	`${ifAlias} like .*EtherChannel.*`
	Match any incident with a CIA attribute value of `Chassis Fan Tray` followed by a digit and Object Identifier (OID) of `.1.3.6.1.4.1.9.9.13.1.4.1.3`
	Note: To include literal strings in the value, enclose the string value in `\Q<literal_value>\E` as shown in the following example.
	`${valueOfCia(\Q.1.3.6.1.4.1.9.9.13.1.4.1.3\E)} like Chassis Fa`

Valid Operator Values, continued

Operator	Description
	`n Tray \d`
not like	Finds all matches that do not have the values specified using the syntax defined for Java regular expressions. For more information,see the Pattern (Java Platform SE6) API documentation at : `http://download.oracle.com/javase/6/docs/api/java/util/regex/Pattern.html` **Note:** To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E . The period asterisk (.*) characters mean *any number of characters of any type at this location*. The period (.) character means *any single character of any type at this location*. Click here for an example. Match any incident with a Source Object's (interface name) ifName value that does not include `rtr`: `${ifName} not like .*rtr.*`

Valid Expressions

Attribute	Description
Expression	The value or pattern for which you want NNMi to search. Note the following: • The expression can include a valid Attribute. • The value or pattern you want to match is case sensitive. • When entering the value for ifSpeed, use the actual numeric value for the interface speed. For example, use `10000000` for ifSpeed **10 Mbps**.

Configure a Source Object Filter for a Causal Rule

The Source Filter tab enables you to create a filter to specify which Source Object should be used for the Parent Incident that is generated for this Causal Rule.

Note: Create only one Source Object Filter for a Causal Rule. If you select **Use Child Incident's Source Object for Parent**☑, NNMi ignores any Source Object Filter you configure.

For information about each Causal Rules tab:

For information about each Child Incident tab:

Use the Filter Editor Buttons to insert Boolean Operators and to append, insert, and replace expressions in the Filter String. Use the Drag and Drop feature to make changes to the placement of the expressions in your Filter String. Click here for more information about using the Filter Editor for Custom Correlations:

- You can use Custom Incident Attributes, attributes for an incident's Source Node or Source Object, or both to define how matching incidents should be considered for the Correlation Rule. See Valid Attributes for more information.
- When specifying Attribute names and values, NNMi uses the type to determine a match. For example, if the Attribute type is numeric, NNMi does a numeric comparison. If the Attribute type is textual, NNMi does a lexographical string comparison. In all cases, when you use the **like** or **not like** operator, NNMi uses a lexographical string comparison. Click here for more information about Attribute types:
  - `ifIndex` and `ifSpeed` are numeric Attributes.
  - Any Attribute name that begins wtih "is" (`isSnmpInterface`, `isSnmpNode`, `isNnmSystemLocal`) represents a Boolean Attribute.
  - All other Attributes are textual.
- Each set of expressions associated with a Boolean Operator (for example, `AND`) is treated as if it were enclosed in parentheses and evaluated together.
- View the expression displayed under **Filter String** to see the logic of the expression as it is created.
- The `AND` and `OR` Boolean Operators must contain at least two expressions.
- The placement of your cursor and the subsequent text that is selected is important when performing operations using the Additional Filters Editor. For example, you append to or replace, the expression that is selected.
- The placement of your cursor and the subsequent text that is selected is especially important when adding your Boolean operators. See "Add Boolean Operators in the Additional Filters Editor" on page 310 for more information.

Filter Editor Buttons and Drag and Drop Feature

Button or Feature	Description
Append	Appends the current expression (Attribute, Operator,and Value) to the selected expression already included in the filter string.
Insert	Inserts the current expression (Attribute, Operator,and Value) in front of the cursor location within the Filter String.
Replace	Replaces the selected expression with the expression displayed Left or Right Expression.
AND	Inserts the AND Boolean Operator in the selected cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.

Filter Editor Buttons and Drag and Drop Feature, continued

Button or Feature	Description
OR	Inserts the OR Boolean Operator in the current cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
Delete	Deletes the selected expression. **Note**: If the Boolean Operator is selected, the Filter Editor deletes all expressions associated with the Boolean Operator.
Drag and Drop	You can drag any of the following items to a new location in the Filter String: ▪ Filter Editor Options: AND, OR, NOT, EXISTS, NOT EXISTS ▪ Filter Expression (Attribute, Operator and Value) When moving items in the Filter String, note the following: ▪ Click the item you want to move before dragging it to a new location. ▪ As you drag a selected item, an underline indicates the target location. ▪ If you are moving the selection up, NNMi places the item above the target location. ▪ If you are moving the selection down, NNMi places the item below the target location. ▪ If you attempt to move the selection to an invalid target location, NNMi displays an error message.

To configure a Source Object Filter for a Causal Rule:

1. Navigate to the **Custom Correlation Configuration** form:
   a. From the workspace navigation pane, select the **Configuration** workspace.
   b. Expand the **Incidents** folder.
   c. Select **Custom Correlation Configuration**.
2. Navigate to the **Causal Rules** tab.
3. From the **Causal Rules** table toolbar, do one of the following:
   - To create a Causal Rule, click the ✳ New icon, and continue.
   - To edit a Causal Rule, click the Open icon in the row representing the Causal Rule you want to edit, and continue.
   - To delete a Causal Rule, click the ✖ Delete icon.
4. Create your Causal Rule. (See "Configure a Causal Rule" on page 714.)

5. Create your Child Incident Configuration . (See "Configure a Child Incident for a Causal Rule" on page 719.)
6. *Optional*. Configure a Child Incident Filter. (See "Configure a Child Incident Filter for a Causal Rule" on page 721.)
7. *Optional*. Configure a Source Object Filter. (See the tables that follow, starting with Filter Editor Components).

   Filter Editor Components

Component	Description
Attribute	The Attribute on which NNMi searches. See Valid Attributes below for a description of valid Attributes.
Operator	Use this Operator to establish the relationship between the Attribute and Expression. See Valid Operators in the table below for the description of each valid Operator.
Expression	Use the Expression to complete the criteria for the Parent Incident configuration. See Valid Expressions below for a description of the components that you might include in your Expression.

8. *Optional*. Configure a Source Node Filter. (See "Configure a Source Node Filter for a Causal Rule" on page 738.)
9. Click **Save and Close** to save your changes and return to the previous form.

Valid Attributes

Attribute	Description
Attribute	The Attribute on which NNMi searches. Valid attributes other than Source Node attributes depend on the Incident's Source Object. NNMi checks the Source Node as well as the Source Object for any Capability value. Note the following when specifying Attributes: • Boolean Attributes begin with "is" and must contain the value `true` or `false`. • If you use attributes that are valid for the Source Node, NNMi uses the Source Node when comparing values. If you use attributes that are valid for the Source Object, NNMi uses the Source Object when comparing values. You cannot use attributes that are valid for the Source Node and Source Object in the same filter. • When using attributes for a Source Object, the attribute must be valid for the incident's Source Object or NNMi does not find a match. For example, if you use the `hostedOn` attribute and the Source Object is not an interface, the correlation does not occur. **Tip**: To check a Source Object for an incident, select the incident of interest, then select Open from the Lookup menu for the Source Object, and examine the Source Object form. A Source Object attribute value of `None` indicates that NNMi cannot identify the

Valid Attributes, continued

Attribute	Description
	Source Object or the Source Object is a Node. If you want to match the incident, use one or more Source Node attributes. • Interface [click here for a list of attribute values] Use the following syntax to specify a Custom Attribute (CA) for an Interface: `valueOfInterfaceCa(<CA_Name>)` For example: `${child.valueOfInterfaceCA(Role)} = WAN Connection` **Values from the Basics Attributes listed on the Interface Form**: ▪ hostedOn (Hosted On Node) **Note**: You must use the full DNS name for the hostedOn value. **Values from the Interface Form: General Tab**: ▪ ifName (name configured for the interface) ▪ ifAlias (alias configured for the interface) ▪ ifDescr (description configured for the interface) ▪ ifIndex (index assigned to the interface) ▪ ifSpeed (speed configured for the interface) **Note**: When entering the value for ifSpeed, use the actual numeric value for the interface speed. For example, use `10000000` for ifSpeed **10 Mbps**. **Addresses from the Interface Form: IP Addresses Tab**: ▪ ipAddress (IP Address associated with the interface) Because NNMi uses a lexicographical compare when evaluating IP addresses, it is recommended that you use the **like** and **not like** operators to specify IP address ranges rather than using the following operators: **>**, **>=**, **<**, or **<=** . **Unique Keys from the Interface Form: Capabilities Tab**: ▪ capability (Unique Key of the Capability) **Values from the Basics Attributes on the parent Node Form**: ▪ isSnmpInterface (Agent Enabled) **Values from the parent Node Form: General Tab**: ▪ sysOidInterface (System Object ID) **Values from the Basics Attributes on the associated Device Profile Form**: ▪ devVendorInterface (Device Vendor) ▪ devFamilyInterface (Device Family) • IP Address [click here for a list of attribute values]

Valid Attributes, continued

Attribute	Description
	Unique Keys from the IP Address Form: Capabilities Tab: ▪ capability (Unique Key of the Capability) • Node [click here for a list of attribute values] Use the following syntax to specify a Custom Attribute (CA) for a Node: `valueOfNodeCa(<CA_Name>)` For example: `${valueOfNodeCa(Location)} = USA` **Values from the Basics Attributes on the Node Form**: ▪ hostname (Hostname, *case-sensitive*) ▪ mgmtIPAddress (Management Address) ▪ isSnmpNode (Agent Enabled) ▪ isNnmSystemLocal (NNMi Management Server) **Values from the Node Form: General Tab**: ▪ sysName (System Name) ▪ sysContact (System Contact) ▪ sysLocation (System Location) ▪ sysOidNode (System Object ID) **Addresses from the Node Form: IP Addresses Tab**: ▪ hostedIPAddress (Address) Because NNMi uses a lexicographical compare when evaluating IP addresses, it is recommended that you use the **like** and **not like** operators to specify IP address ranges rather than using the following operators: **>**, **>=**, **<**, or **<=** . **Unique Keys from the Node Form: Capabilities Tab**: ▪ capability (Unique Key of the Capability) **Values from the Basics Attributes on the associated Device Profile Form**: ▪ devVendorNode (Device Vendor) ▪ devFamillyNode (Device Family) **Values from the associated entry on the Regional Manager Form: Connection Tab**: ▪ nnmSystemName (Hostname, *case-sensitive*) (*NNMi Advanced*) If the Global Network Management feature is enabled, this attribute value identifies a Regional Manager (NNMi management server).

Valid Operator Values

Operator	Description
=	Finds all values equal to the value specified. Click here for examples. Match any incident with a CIA value of 5 and Object Identifier (OID) of .1.3.6.1.4.1.9.9.106.2.0.1: `${valueOfCia(\Q.1.3.6.1.4.1.9.9.106.2.0.1\E)} = 5`
!=	Finds all values not equal to the value specified. Click here for an example. Match any incident with a Source Object value of Interface with Device Vendor value not equal to `Cisco`: `${devVendorInterface} != Cisco`
<	Finds all values less than the value specified. Click here for an example. Match any incident with a CIA value less than 5 and Object Identifier (OID) value of .1.3.6.1.4.1.9.9.106.2.0.1: `${valueOfCia(\Q.1.3.6.1.4.1.9.9.106.2.0.1\E)} < 5`
<=	Finds all values less than or equal to the value specified. Click here for examples. Match any incident with a CIA attribute value less than or equal to 5 and Object Identifier (OID) value of .1.3.6.1.4.1.9.9.106.2.0.1: `${valueOfCia (\Q.1.3.6.1.4.1.9.9.106.2.0.1\E)} <= 5`
>	Finds all values greater than the value specified. Click here for an example. Match any incident with a CIA value greater than 5 and Object Identifier (OID) attribute value of .1.3.6.1.4.1.9.9.106.2.0.1: ${valueOfCia (\Q.1.3.6.1.4.1.9.9.106.2.0.1\E)} > 5
>=	Finds all values greater than or equal to the value specified. Click here for an example. Match any incident with a Source Object attribute value of Interface that has an (interface speed) ifSpeed of 10Mbps: `${ifSpeed} >= 10000000`
is not null	Finds all non-blank values. Click here for an example. Match any incident with a Source Object attribute value of Interface that has an

Valid Operator Values, continued

Operator	Description
	(interface name) ifName value: `${ifName} is not null`
is null	Finds all blank values. Click here for an example. Match any incident with a Source Object attribute value of Interface that does not have an (interface name) ifName value: `${ifName} is null`

like	Finds matches using wildcard characters and the question mark.
	The asterisk (*) character means *any number of characters of any type at this location*.
	The question mark (?) character means *any single character of any type at this location*.
	Click here for an example.
	Match any incident with a Source Object attribute value of Interface with a (description) ifDescr that begins with `Serial` followed by any number of characters:
	`${ifDescr} like Serial*`
	Match any incident with a Source Object attribute value of Interface with an (interface alias) ifAlias value that begins with `EtherChannel` (for example, `EtherChannel Group 1`).
	`${ifAlias} like EtherChannel*`

not like	Finds all matches that do not have the values specified. The asterisk (*) characters means *any number of characters of any type at this location*. The question mark (?) character means *any single character of any type at this location*. Click here for an example. Match any with a Source Object attribute value of Interface with an (interface name) ifName value that does not begin with `rtr*`: `${ifName} not like rtr*`

Valid Expressions

Attribute	Description
Expression	The value or pattern for which you want NNMi to search. Note the following:

Valid Expressions, continued

Attribute	Description
	• The expression can include a valid Attribute. • The value or pattern you want to match is case sensitive. • When entering the value for ifSpeed, use the actual numeric value for the interface speed. For example, use `10000000` for ifSpeed **10 Mbps**.

Configure a Source Node Filter for a Causal Rule

The Source Node Filter tab enables you to create a filter to specify which Source Node should be used for the Parent Incident that is generated for this Causal Rule.

Note: Create only one Source Node Filter for a Causal Rule. If you select **Use Child Incident's Source Node for Parent**☑, NNMi ignores any Source Node Filter you configure.

For information about each Causal Rules tab:

For information about each Child Incident tab:

Use the Filter Editor Buttons to insert Boolean Operators and to append, insert, and replace expressions in the Filter String. Use the Drag and Drop feature to make changes to the placement of the expressions in your Filter String. Click here for more information about using the Filter Editor:

- You can use Custom Incident Attributes, attributes for an incident's Source Node, or both to define how matching incidents should be considered for the Causal Rule. See Valid Attributes for more information.
- When specifying Attribute names and values, NNMi uses the type to determine a match. For example, if the Attribute type is Integer, NNMi does a numeric comparison. If the Attribute type is textual, NNMi does a lexographical string comparison. In all cases, when you use the **like** or **not like** operator, NNMi uses a lexographical string comparison.
- Each set of expressions associated with a Boolean Operator is treated as if it were enclosed in parentheses and evaluated together.
- View the expression displayed under **Filter String** to see the logic of the expression as it is created.
- The `AND` and `OR` Boolean Operators must contain at least two expressions.
- The placement of your cursor and the subsequent text that is selected is important when performing operations using the Additional Filters Editor. For example, you append to or replace, the expression that is selected.
- The placement of your cursor and the subsequent text that is selected is especially important when adding your Boolean operators. See "Add Boolean Operators in the Additional Filters Editor" on page 310 for more information.

Filter Editor Buttons and Drag and Drop Feature

Button or Feature	Description
Append	Appends the current expression (Attribute, Operator,and Value) to the selected expression already included in the filter string.
Insert	Inserts the current expression (Attribute, Operator,and Value) in front of the cursor location within the Filter String.
Replace	Replaces the selected expression with the expression displayed Left or Right Expression.
AND	Inserts the AND Boolean Operator in the selected cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
OR	Inserts the OR Boolean Operator in the current cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
Delete	Deletes the selected expression. **Note**: If the Boolean Operator is selected, the Filter Editor deletes all expressions associated with the Boolean Operator.
Drag and Drop	You can drag any of the following items to a new location in the Filter String: ▪ Filter Editor Options: AND, OR, NOT, EXISTS, NOT EXISTS ▪ Filter Expression (Attribute, Operator and Value) When moving items in the Filter String, note the following: ▪ Click the item you want to move before dragging it to a new location. ▪ As you drag a selected item, an underline indicates the target location. ▪ If you are moving the selection up, NNMi places the item above the target location. ▪ If you are moving the selection down, NNMi places the item below the target location. ▪ If you attempt to move the selection to an invalid target location, NNMi displays an error message.

To configure a Source Node Filter for a Causal Rule:

1. Navigate to the **Custom Correlation Configuration** form:

   a. From the workspace navigation pane, select the **Configuration** workspace.

   b. Expand the **Incidents** folder.

   c. Select **Custom Correlation Configuration**.

2. Navigate to the **Causal Rules** tab.

3. From the **Causal Rules** table toolbar, do one of the following:

   - To create a Causal Rule, click the New icon, and continue.
   - To edit a Causal Rule, click the Open icon in the row representing the Causal Rule you want to edit, and continue.
   - To delete a Causal Rule, click the Delete icon.

4. Create your Causal Rule. (See "Configure a Causal Rule" on page 714.)

5. Create your Child Incident Configuration . (See "Configure a Child Incident for a Causal Rule" on page 719.)

6. *Optional*. Configure a Child Incident Filter. (See "Configure a Child Incident Filter for a Causal Rule" on page 721.)

7. *Optional*. Configure a Source Object Filter. (See "Configure a Source Object Filter for a Causal Rule" on page 730.)

8. *Optional*. Configure a Source Node Filter. (See the tables that follow, starting with Filter Editor Components.)

   Filter Editor Components

Component	Description
Attribute	The Attribute on which NNMi searches. See Valid Attributes below for a description of valid Attributes.
Operator	Use this Operator to establish the relationship between the Attribute and Expression. See Valid Operators in the table below for the description of each valid Operator.
Expression	Use the Expression to complete the criteria for the Parent Incident configuration. See Valid Expressions below for a description of the components that you might include in your Expression.

9. Click **Save and Close** to save your changes and return to the previous form.

Valid Attributes

Attribute	Description
Attribute	The Attribute on which NNMi searches. Note the following when specifying Attributes: • Boolean Attributes begin with "is" and must contain the value `true` or `false`. • If you use attributes that are valid for the Source Node, NNMi uses the Source Node when comparing values. If you use attributes that are valid for the Source Object, NNMi uses the Source Object when comparing values. You cannot use attributes that are valid for the Source Node and Source Object in the same filter. ▪ When using attributes for a Source Object, the attribute must be valid for the

Valid Attributes, continued

Attribute	Description
	incident's Source Object or NNMi does not find a match. For example, if you use the `hostedOn` attribute and the Source Object is not an interface, the correlation does not occur. **Tip**: To check a Source Object for an incident, select the incident of interest, then select Open from the Lookup menu for the Source Object, and examine the Source Object form. A Source Object attribute value of `None` indicates that NNMi cannot identify the Source Object or the Source Object is a Node. If you want to match the incident, use one or more Source Node attributes. • Node [click here for a list of attribute values] Use the following syntax to specify a Custom Attribute (CA) for a Node: `valueOfNodeCa(<CA_Name>)` For example: `${valueOfNodeCa(Location)} = USA` **Values from the Basics Attributes on the Node Form**: ▪ hostname (Hostname, *case-sensitive*) ▪ mgmtIPAddress (Management Address) ▪ isSnmpNode (Agent Enabled) ▪ isNnmSystemLocal (NNMi Management Server) **Values from the Node Form: General Tab**: ▪ sysName (System Name) ▪ sysContact (System Contact) ▪ sysLocation (System Location) ▪ sysOidNode (System Object ID) **Addresses from the Node Form: IP Addresses Tab**: ▪ hostedIPAddress (Address) Because NNMi uses a lexicographical compare when evaluating IP addresses, it is recommended that you use the **like** and **not like** operators to specify IP address ranges rather than using the following operators: **>**, **>=**, **<**, or **<=**. **Unique Keys from the Node Form: Capabilities Tab**: ▪ capability (Unique Key of the Capability) **Values from the Basics Attributes on the associated Device Profile Form**: ▪ devVendorNode (Device Vendor) ▪ devFamillyNode (Device Family) **Values from the associated entry on the Regional Manager Form:**

Valid Attributes, continued

Attribute	Description
	Connection Tab: - nnmSystemName (Hostname, *case-sensitive*) (*NNMi Advanced*) If the Global Network Management feature is enabled, this attribute value identifies a Regional Manager (NNMi management server).

Valid Operator Values

Operator	Description
=	Finds all values equal to the value specified. Click here for examples. Match any incident with a CIA value of 5 and Object Identifier (OID) value of .1.3.6.1.4.1.9.9.106.2.0.1: `${valueOfCia(\Q.1.3.6.1.4.1.9.9.106.2.0.1\E)} = 5`
!=	Finds all values not equal to the value specified. Click here for an example. Match any incident with a Source Node value that has a Device Vendor value not equal to `Cisco`: `${devVendorNode} != Cisco`
<	Finds all values less than the value specified. Click here for an example. Match any incident with a CIA value less than 5 and Object Identifier (OID) value of .1.3.6.1.4.1.9.9.106.2.0.1: `${valueOfCia(\Q.1.3.6.1.4.1.9.9.106.2.0.1\E)} < 5`
<=	Finds all values less than or equal to the value specified. Click here for examples. Match any incident with a CIA value less than or equal to 5 and Object Identifier (OID) value of .1.3.6.1.4.1.9.9.106.2.0.1: `${valueOfCia (\Q.1.3.6.1.4.1.9.9.106.2.0.1\E)} <= 5`
>	Finds all values greater than the value specified. Click here for an example. Match any incident with a CIA value greater than 5 and Object Identifier (OID) value of .1.3.6.1.4.1.9.9.106.2.0.1: ${valueOfCia (\Q.1.3.6.1.4.1.9.9.106.2.0.1\E)} > 5

Valid Operator Values, continued

Operator	Description
>=	Finds all values greater than or equal to the value specified.
is not null	Finds all non-blank values. Click here for an example. Match any incident with a Source Node that has a (system contact name) sysContact value: `${sysContact} is not null`
is null	Finds all blank values. Click here for an example. Match any incident with a Source Node that does not have a (system contact name) sysContact value: `${sysContact} is null`
like	Finds matches using wildcard characters and the question mark. The asterisk (*) character means *any number of characters of any type at this location*. The question mark (?) character means *any single character of any type at this location*. Click here for an example. Match any incident with a Source Node that has a (system location) sysLocation value that begins with Bldg5: `${syslocation} like Bldg5*`
not like	Finds all matches that do not have the values specified. The asterisk (*) characters means *any number of characters of any type at this location*. The question mark (?) character means *any single character of any type at this location*. Click here for an example. Match any incident with a Source Node that has a (system location) sysLocation value that does not begin with `Bldg5`: `${sysLocation} not like Bldg5*`

Valid Expressions

Attribute	Description
Expression	The value or pattern for which you want NNMi to search.

Valid Expressions, continued

Attribute	Description
	Note the following: • The expression can include a valid Attribute. • The value or pattern you want to match is case sensitive.

Causal Rule Example

Tip: Use these steps as a guideline for creating your own Causal Rules.

This example creates a Causal Rule that generates a new CardHealthProblem Parent Incident. It uses the traps described in the following table to determine the following:

• Whether there is a temperature problem or diagnostic failure for a Field Replaceable Unit (FRU) Card module

• Whether the source of the problem is a fan, a power supply, or both.

Trap Descriptions

Trap	Description
FruModuleStatusChange	Indicates a temperature problem (14) or diagnostic failure (11) for the Field Replaceable Unit (FRU) card module
CiscoEnvMonFanNotification	Indicates the problem is related to a fan. The example Causal Rule uses this trap to obtain the name of the fan.
CiscoEnvMonSuppStatusChangeNotif	Indicates the problem is related to the Power Supply.

Using the Causal Rule described in this example, NNMi generates a new CardHealthProblem Parent Incident when NNMi determines the following:

- The Source Object for the Child Incident is a Field Replaceable Unit (FRU) card.

  Note: NNMi checks for the **com.hp.nnm.capability.card.fru** capability to determine whether the Source Object is an FRU card.

- The FruModuleStatusChange trap returns a value of either 14 (temperature problem) or 11 (diagnostic failure).

To configure the CardHealth Causal Rule Basics information:

1. Navigate to the **Causal Rule** form:

   a. From the workspace navigation pane, select the **Configuration** workspace.

   b. Expand the **Incidents** folder.

   c. Select **Custom Correlation Configuration**.

2. Navigate to the **Causal Rules** tab.

3. From the **Causal Rules** table toolbar, click the ✱ New icon.
4. In the **Name** attribute, enter a unique name that will help you to identify the Causal Rule. In this example, the Causal Rule Name is **Card Health**.
5. In the **Author** attribute, either enter a name that identifies the person who is creating the Causal Rule or keep the default value **Customer**.
6. Make sure **Enabled** ☑ is checked to indicate the NNMi Causal Engine should use this Causal Rule when evaluating incidents.
7. To create a new Incident Configuration for the Parent Incident, in the **Parent Incident** Lookup Field, select ✱ New.
8. In the Management Event Configuration form, enter the **Basics** information as follows:
   a. In the **Name** attribute, enter **CardHealthProblem** for the Name value.
   b. Make sure **Enabled** ☑ is checked to indicate the NNMi Causal Engine should use this Causal Rule when evaluating incidents.
   c. In the **Categories** Lookup Field, select Quick Find and select **Fault** from the list of incident Categories.
   d. In the **Family** Lookup Field, select Quick Find and then **Card** from the list of incident Families.
   e. In the **Severity** Lookup Field, select Quick Find and then **Critical** from the list of incident Severities.
   f. In the Message Format attribute, enter the following: `Card $.1.3.6.1.2.1.47.1.1.1.1.7.5000 with $.1.3.6.1.4.1.9.9.13.1.4.1.2 and Power Supply not functioning`

      NNMi displays the name of the Card using the Object Identifier (OID) value of $.1.3.6.1.2.1.47.1.1.1.1.7.5000. NNMi displays the name of the Fan using the OID value of $.1.3.6.1.4.1.9.9.13.1.4.1.2.
   g. Click **Save and Close** to save your changes and return to the **Causal Rule** form.
9. In the **Correlation Nature** select **Root Cause** from the drop-down list.
10. In **Common Child Incident Attribute**, enter **${hostname}**.
11. In the **Correlation Window Duration** attribute, keep the default value of **5** minutes.
12. Use the **Description** attribute to provide additional information that you would like to store about the current incident configuration. This description applies only to the configuration entry.

To configure the first Child Incident (CiscoModuleStatusChange):

1. In the **Causal Rule** form, navigate to the **Child Incidents** tab.
2. Click the ✱ New icon to configure the first Child Incident.
3. In the **Name** attribute of the Child Incident Configuration form, enter **FRU Card**.
4. In the **Child Incident** Lookup Field, select Quick Find and then **CiscoModuleStatusChange** from the list of incident configurations.

5. To forward the Card name to the new Parent Incident, in **Forward Child Custom Incident Attributes**, enter **.1.3.6.1.2.1.47.1.1.1.1.7.5000**.
6. Check to enable **Use Child Incident's Source Object for Parent** ☑.
7. Check to enable **Use Child Incident's Source Node for Parent** ☑.

To configure the first Child Incident Filter:

1. In the **Child Incident Configuration** form, navigate to the **Child Incident Filter** tab.

   Next, create the following filter: `(capability = com.hp.nnm.capability.card.fru AND ${valueOfCia(\Q.1.3.6.1.4.1.9.9.117.1.2.1.1.2.5000\E)} = 11) OR ${valueOfCia(\Q.1.3.6.1.4.1.9.9.117.1.2.1.1.2.5000\E)} = 14)`
2. In the **Attribute** field, enter **capability**.
3. In the **Operator** field, select **=** from the drop-down menu.
4. In the **Expression** field, enter **com.hp.nnm.capability.card.fru**.
5. Click **Append**.
6. Select **Insert** from the drop-down menu.
7. Click **AND**.
8. Click to select **AND** in the Child Incident Filter Expression.
9. Select **Append** from the drop-down menu.
10. Click **OR**.
11. Click to select **OR** in the Child Incident Filter Expression.
12. In the **Attribute** field, enter **${valueOfCia(\Q.1.3.6.1.4.1.9.9.117.1.2.1.1.2.5000\E)}**.
13. In the **Operator** field, select **=** from the drop-down menu.
14. In the **Expression** field, enter **11**.
15. Click **Append**.
16. In the **Attribute** field, enter **${valueOfCia(\Q.1.3.6.1.4.1.9.9.117.1.2.1.1.2.5000\E)}**.
17. In the **Operator** field, select **=** from the drop-down menu.
18. In the **Expression** field, enter **14**.
19. Click to select **OR** in the Child Incident Filter Expression.
20. Click **Append**.
21. Click **Save and Close** to return to the **Causal Rule** form.

To configure the second Child Incident (CiscoEnvMonFanNotification):

1. In the **Causal Rule** form, navigate to the **Child Incidents** tab.
2. Click the ✳ New icon to configure the second Child Incident.
3. In the **Name** attribute of the **Child Incident Configuration** form, enter **Chassis Fan**.

4. In the **Child Incident** Lookup Field, select Quick Find and then **CiscoEnvMonFanNotification** from the list of incident configurations.
5. To forward the Fan name to the new Parent Incident, in **Forward Child Custom Incident Attributes**, enter **.1.3.6.1.2.1.47.1.1.1.1.7.5000**.

To configure the second Child Incident Filter:

1. In the **Child Incident Configuration** form, navigate to the **Child Incident Filter** tab.

   Next, create the following filter: `(${valueOfCia(\Q.1.3.6.1.4.1.9.9.13.1.4.1.2\E)} = Chassis Fan Tray 1 AND ${valueOfCia(\Q.1.3.6.1.4.1.9.9.13.1.4.1.3\E)} = 3)`
2. In the **Attribute** field, enter **${valueOfCia(\Q.1.3.6.1.4.1.9.9.13.1.4.1.2\E)}**.
3. In the **Operator** field, select **=** from the drop-down menu.
4. In the **Expression** field, enter **Chassis Fan Tray 1**.
5. Click **Append**.
6. Select **Insert** from the drop-down menu.
7. Click **AND**.
8. In the **Attribute** field, enter **${valueOfCia(\Q.1.3.6.1.4.1.9.9.13.1.4.1.3\E)}**.
9. In the **Operator** field, select **=** from the drop-down menu.
10. In the **Expression** field, enter **3**.
11. Click **Append**.
12. Click **Save and Close** to return to the **Causal Rule** form.

To configure the third Child Incident (CiscoEnvMonSuppStatusChangeNotif):

1. In the **Causal Rule** form, navigate to the **Child Incidents** tab.
2. Click the New icon to configure the third Child Incident.
3. In the **Name** attribute of the **Child Incident Configuration** form, enter **Chassis Power**.
4. In the **Child Incident** Lookup Field, select Quick Find and then **CiscoEnvMonSuppStatusChangeNotif** from the list of incident configurations.
5. To forward the Fan name to the new Parent Incident, in **Forward Child Custom Incident Attributes**, **enter ${valueOfCia(\Q.1.3.6.1.4.1.9.9.13.1.4.1.2\E)}**.

To configure the third Child Incident Filter:

1. In the **Child Incident Configuration** form, navigate to the **Child Incident Filter** tab.

   Next, create the following filter: `(${valueOfCia(\Q.1.3.6.1.4.1.9.9.13.1.4.1.3\E} = 3} AND ${valueOfCia(\Q.1.3.6.1.4.1.9.9.117.1.2.1.1.2.5000\E} = Power Supply 1, WS-CAC-1300W)`
2. In the **Attribute** field, enter **${valueOfCia(\Q.1.3.6.1.4.1.9.9.13.1.4.1.3\E)}**.
3. In the **Operator** field, select **=** from the drop-down menu.

4. In the **Expression** field, enter **3**.
5. Click **Append**.
6. Select **Insert** from the drop-down menu.
7. Click **AND**.
8. In the **Attribute** field, enter **${valueOfCia(\Q.1.3.6.1.4.1.9.9.13.1.4.1.2\E)}**.
9. In the **Operator** field, select **=** from the drop-down menu.
10. In the **Expression** field, enter **Power Supply 1, WS-CAC-1300W**.
11. Click **Append**.
12. Click **Save and Close** to save your changes and return to the **Causal Rule** form.
13. Click **Save and Close** to save your changes and return to the **Custom Correlation Configuration** form.
14. Click **Save and Close** to save the Custom Correlation Configuration.

See "Correlation Rule Example" on page 710 for an example of creating a Correlation Rule.

Configure an Action for an Incident

You can configure actions to automatically run at any point in the incident lifecycle. For example, you might want to configure an action to occur when an incident of the type you are configuring is generated (**Registered**). When an incident is generated, you might want to automatically open a trouble ticket or send email or page your network operator. After the incident is **Closed**, you might want to automatically close the trouble ticket.

> **Note:** Your actions will not be executed until you enable the Actions configuration by either clicking Enable ☑ on the Actions tab or using the **Actions** → **Enable Configuration** option.

If the NNMi management server is running on a Windows operating system, NNMi runs each action that you configure using the Local System account. If the NNMi management server is running on an HP-UX, Solaris or Linux operating system, NNMi runs each action that you configure using the bin user name. To change the user account associated with actions, see the "Setting the Action Server Name Parameter" section in the *HP Network Node Manager i Software Deployment Reference*.

You can provide the required information within the following contexts:

"Configure Actions for an SNMP Trap Incident" on page 913

"Configure Actions for a Remote NNM 6.x/7.x Event Incident" on page 1357

"Configure Actions for a Management Event Incident" on page 1206

Lifecycle Transition Action Form

Use this form to enter the command you want to run when an incident of the type you are configuring is at a particular Lifecycle State. For example, when an incident is generated

(**Registered**), you might want to automatically open a trouble ticket or email or page your network operator.

You can provide the required information within the following contexts:

"Lifecycle Transition Action Form (SNMP Trap Incidents)" on page 914

"Lifecycle Transition Action Form (Remote NNM 6.x/7.x Events)" on page 1358

"Lifecycle Transition Action Form (Management Events)" on page 1207

Valid Parameters for Configuring Incident Actions (Management Events)

When configuring incident actions, consider using incident information as part of the action. NNMi provides the following parameter values. Use these parameters as variables in your Jython or executable files.

Tip: See the Using the Incident Form for more information about the parameter values.

Note: NNMi stores varbind values as custom incident attributes (CIAs).

Tip: If a value is not stored for a parameter, it is returned as “null”.

See "Lifecycle Transition Action Form" on the previous page for more information about configuring incident actions.

Valid Parameters Visible From an Incident's Form

Parameter Value	Description
$category, $cat	Value of the Category attribute in the Incident form.
$count, $cnt	Value representing the number of Custom Incident Attributes that appear in the Incident form.
$family, $fam	Value from the Family attribute in the Incident form.
$firstOccurrenceTime, $fot	Value from the First Occurrence Time attribute in the incident form.
$lastOccurrenceTime, $lot	Value from the Last Occurrence Time attribute in the incident form.
$lifecycleState, $lcs	Value from the Lifecycle State attribute in the Incident form.
$name	Value of the Name attribute from the incident configuration.
$nature, $nat	Value from the Nature attribute in the Incident form.
$origin, $ori	Value from the Origin attribute in the Incident form.
$originOccurrenceTime, $oot	Value from the Origin Occurrence Time attribute in the incident form.
$priority, $pri	Value from the Priority attribute in the Incident form.
$severity, $sev	Value of the Severity attribute of the Incident form.

Valid Parameters Visible from a Node Form

Parameter Value	Description
$managementAddress, $mga	Value from the Management Address attribute of the incident's source Node's form or SNMP Agent form.
$otherSideOfConnectionManagementAddress, $oma	If the incident's Source Node is part of a Layer 2 Connection, this attribute is the value of the Management Address of a node on the other side of the Layer 2 Connection.
$sourceNodeLongName, $sln	The fully-qualified DNS name as displayed in the Hostname attribute of the incident's source Node's form.
$sourceNodeName, $snn	Value from the Name attribute of the incident's source Node's form.
$sysContact, $sct	Value from the System Contact attribute of the incident's source Node form: General tab.
$sysLocation, $slc	Value from the System Location attribute of the incident's source Node form: General tab.

Valid Parameters Visible from an Interface Form

Parameter Value	Description
$ifAlias, $ifa	Value from the IfAlias attribute for the interface that is the incident's source object.
$ifConfigDupSetting, $icd	Configured Duplex Setting on the port associated with the interface that is the incident's source object.
$ifDesc, $idc	Value from the ifDesc attribute for the interface that is the incident's source object.
$ifIndex, $idx	Value from the ifIndex attribute for the interface that is the incident's source object.
$ifIpAddr, $iia	IP Address values associated with the interface that is the incident's source object. If multiple IPaddresses are associated with the interface, this parameter returns a comma-separated list.
$ifName, $ifn	Value from the ifName attribute for the interface that is the incident's source object.
$ifPhysAddr, $ipa	Value from the Physical Address attribute for the interface that is the incident's source object.

Valid Parameters Visible from an Interface Form , continued

$ifSpeed, $isp	Value from the ifSpeed attribute for the interface that is the incident's souce object.
$ifType, $itp	Value from the ifType attribute for the interface that is the incident's souce object.

Valid Parameters Visible from a Layer 2 Connection Form

Parameter Value	Description
$otherSideOfConnectionConfigDupSetting, $ocd	If the incident's source Node is part of a Layer 2 Connection, this parameter contains the Configured Duplex Setting on the port associated with the interface on the other side of the connection.
$otherSideOfConnectionIfAlias, $oia	If the incident's Source Node is part of a Layer 2 Connection, this parameter is the value of the ifAlias of one of the interfaces on the other side of the Layer 2 Connection.
$otherSideOfConnectionIfDesc, $odc	If the incident's Source Node is part of a Layer 2 Connection, this parameter contains the ifDescr attribute value for the interface on the other side of the Layer 2 Connection.
$otherSideOfConnectionIfIndex, $odx	If the incident's Source Node is part of a Layer 2 Connection, this parameter contains the ifIndex attribute value for the interface on the other side of the connection.
$otherSideOfConnectionIfName, $ofn	If the incident's Source Node is part of a Layer 2 Connection, this parameter contains the ifName attribute value for the interface on the other side of the connection.

Valid Parameters Visible from a VLAN Form

Parameter Value	Description
$impVlanIds, $ivi	Value from the VLAN Id attribute associated with the interface that is the incident's source object. To access this information from an interface form, navigate to the VLAN Port tab and open the form for the VLAN of interest. If the interface is part of more than one VLAN, this parameter returns a comma-separated list.
$impVlanNames, $ivn	Value from the Global VLAN Name attribute associated with the interface that is the incident's source object. To access this information from a Node form or Interface form, navigate to the VLAN Ports tab. If the node or

Valid Parameters Visible from a VLAN Form, continued

Parameter Value	Description
	interface is part of more than one VLAN, this parameter returns a comma-separated list.

Valid Parameters Not Visible From a Form

Parameter Value	Description
$id	Unique Object Identifier attribute value for the incident (unique across the entire NNMi Database).
$firstOccurrenceTimeMs, $fms	Value from the First Occurrence Time attribute in the incident form, converted to millseconds (measured since January 1, 1970, 00:00:00 GMT - Greenwich Mean Time).
$lastOccurrenceTimeMs, $lms	Value from the Last Occurrence Time attribute in the incident form, converted to millseconds (measured since January 1, 1970, 00:00:00 GMT - Greenwich Mean Time).
$messageFormat, $msg	*Valid for Incident actions only*. Message text displayed for an incident when this parameter is included as an argument to an incident action.
$oid	Value of the unique object identifier (oid) for the incident configuration that originated from either an SNMP trap, Management Event, or Remote NNM 6.x or 7.x event.
$otherSideOfConnection, $osc	If the incident's Source Node is part of a Layer 2 Connection, this attribute is the following combination of values for the node and one of its interfaces on the other side of the Layer 2 Connection: The fully-qualified DNS name of the node appended with the interface Name in the following format: <*fully-qualified DNS name*> [*interface_name*]
$originOccurrenceTimeMs, $oms	Value from the Origin Occurrence Time attribute in the incident form, converted to millseconds (measured since January 1, 1970, 00:00:00 GMT - Greenwich Mean Time).
$sourceNodeUuid, $snu	Universally Unique Object Identifier attribute value of the source node object for the incident (unique across all databases). This identifier distinguishes the source node object instance from all other node objects.
$sourceObjectClass, $soc	Value of the object class for the object you want to include. Use this parameter to request more details of a class of objects through a web service. Examples of object classes include: `com.hp.ov.nms.model.core.Interface` and `com.hp.ov.nms.model.snmp.SnmpAgent`.

Valid Parameters Not Visible From a Form, continued

Parameter Value	Description
$sourceObjectName, $son	Value from the Name attribute of the source object. For example, an interface object is named according to the MIB ifName. Each ifName varies according to the vendor's conventions. Using the name `4/1` as an example, `4` represents `the board number and 1` represents the port number.
$sourceObjectUuid, $sou	Universally Unique Object Identifier attribute value of the source object for the incident (unique across all databases). This identifier distinguishes the source object instance from all other similar object instances..
$uuid	Universally Unique Object Identifier attribute value of the incident (unique across all databases). This identifier distinguishes the incident object instance from all other incident objects.

Valid Parameters Established in Custom Incident Attributes

Parameter Value	Description
$<position_ number>	Value of the custom incident attribute (CIA) position number for any CIA that originated from a varbind or was added by NNMi. For example, to indicate you want to use the varbind in position 1, enter: `$1` NNMi stores varbind values as Custom Incident Attributes. If you know the varbind position number, use this parameter.
$<CIA_ name>	Value of the name that is used for the custom incident attribute. For example, `$mycompany.mycia`. NNMi provides CIA values for configuring Management Events. See Custom Incident Attributes Provided by NNMi for more information about custom incident attributes.
$<CIA_ oid>	Value of the object identifier for any custom incident attribute that originated as a varbind. For example, `$.1.3.6.1.6.3.1.1.5.1`. Use this parameter when you are not certain of a custom incident attribute (varbind) position number.
$*	Used to indicate you want all of the custom incident attribute values originating as varbinds, to be passed to the action configuration. Each varbind is returned in the following format: `$<`*`CIA_name`*`>:<`*`CIA_value`*`>` in which the custom incident attribute name appears followed by the custom incident attribute value.

The function described in the following table replaces the specified numeric value with the associated text value stored in the CIA.

Note: The associated MIB must have been loaded using the nnmloadmib.ovpl command.

Functions to Generate Values Within Incident Messages

Function	Description
$text ($<position_ number>)	The <*position_number*> argument specifies the numeric value of the custom incident attribute (CIA) position number for any CIA that originated from a varbind or was added by NNMi. For example, to indicate you want to use the varbind in position 1, enter: `$1.` After the function runs, NNMi replaces the numeric value with the text value stored in the CIA. **Note**: If a text value is not available, NNMi returns the numeric value.
$text ($<CIA_ oid>)	The <CIA_oid> argument specifies the object identifier for any custom incident attribute that originated as a varbind. For example, `$.1.3.6.1.6.3.1.1.5.1.` Use this argument to the $text function when you are not certain of a custom incident attribute (varbind) position number. After the function runs, NNMi replaces the numeric value with the text value stored in the CIA. **Note**: If a text value is not available, NNMi returns the numeric value.

Handling Special Characters in Action Arguments

In some cases, NNMi requires or inserts double quotes or escape characters in arguments that are passed to the Jython file, executable, or shell script using the **Command** attribute.

Note: Shell commands are not permitted in the **Command** attribute. If you use shell commands, place them in a shell script file and reference that file from the **Command** attribute.

The following table describes how to handle special characters included as arguments to your Jython files, executables, or shell scripts.

Handling Special Characters in Arguments

Circumstance	Result
If the following special characters are requested as a single argument to a Jython, executable, shell script, or shell command: , ; & > < (space) | =	The argument (containing the special character) must be wrapped in double quotes. For example, **"Hello;World"**.
Request all available CIA name/value pairs for a particular incident $*	The $* argument returns a parsed string. For this example, the available CIA name/value pairs are: • **$1** = 123 • **$com.mycompany.mycia** = 012345 • **$.1.3.6.1.2.1.2.2.1.1** = 1007 **Example Command**

Handling Special Characters in Arguments, continued

Circumstance	Result
	`echoScript.bat $*` NNMi returns the following string in response to the command: • **Windows:** `"1:123,com.mycompany.mycia:012345, .1.3.6.1.2.1.2.2.1.1:1007"` • **UNIX:** `1:123,com.mycompany.mycia:012345, .1.3.6.1.2.1.2.2.1.1:1007`
Request specific CIA values as an argument to an action command $<*CIA name, position, or OID*>	To request specific CIA values, use the $ followed by the CIA name **Example Command** `echoScript.bat $1  $com.mycompany.mycia $.1.3.6.1.2.1.2.2.1.1` For this example, the CIA name/value pairs are: • **$1** = 123 • **$com.mycompany.mycia** = 012345 • **$.1.3.6.1.2.1.2.2.1.1** = 1007 NNMi returns the following string in response to the command: • **Windows:** `123   012345   1007` • **UNIX:** `123   012345   1007`

If an invalid CIA name, position, or OID is requested as an argument to an action command	If the trap or event does not contain one or more of the requested CIAs, NNMi passes error messages as arguments.
	UNIX:
	`Invalid or unknown cia position 1`
	`Invalid or unknown cia com.mycompany.mycia`
	`Invalid or unknown cia .1.3.6.1.2.1.2.2.1.1`
	Windows: NNMi encloses each CIA value in double quotes.
	`Invalid or unknown cia "position 1"`
	`Invalid or unknown cia "com.mycompany.mycia"`
	`Invalid or unknown cia ".1.3.6.1.2.1.2.2.1.1"`

Use $* in your incident action	**UNIX:**

Handling Special Characters in Arguments, continued

Circumstance	Result
scripts	It is recommended that you do not use $* (shell variable substitution) in your incident action scripts. If you do use $* within the shell script, specifying $* expands into the arguments and are rescanned. This means that blanks in arguments will result in multiple arguments. If you want to use shell variable substitution, use the `"$@"` instead so that blanks in arguments are ignored.
Use arguments to Jython methods	Enclose any argument that is not preceded with a "$" (dollar sign) in double quotes. For example, jythonMethod($Severity, "Hello; World").

Example Jython Methods Provided by NNMi

NNMi provides a set of example Jython methods you can use when configuring actions for incidents. These example files reside in the following directory:

Windows:

`<drive>\Program Files(x86)\HP\HP BTO Software\newconfig\HPOvNmsEvent/actions`

UNIX:

`/opt/OV/newconfig/HPOvNmsEvent/actions`

If you want to use one or more of these example Jython methods, you must first copy the example files to the following directory :

Windows:

<drive>:`\ProgramData\HP\HP BTO Software\shared\nnm\actions`

<*drive*> is the drive on which NNMi is installed.

UNIX:

`/var/opt/OV/shared/nnm/actions`

See "Lifecycle Transition Action Form" on page 748 for more information about creating incident actions.

Note: The argument values, such as *arg1*, and *arg2*, can be any valid parameter as described in "Valid Parameters for Configuring Incident Actions (Management Events)" on page 1216.

Example Jython Methods Provided by NNMi

File Name	Method	Description
testPrint.py	testPrint_ Registered()	Displays the incident Lifecycle State specified by the method name.

Example Jython Methods Provided by NNMi, continued

File Name	Method	Description
testPrint.py	testPrint_ InProgress()	Displays the incident Lifecycle State specified by the method name.
testPrint.py	testPrint_ Completed()	Displays the incident Lifecycle State specified by the method name.
testPrint.py	testPrint_Closed ()	Displays the incident Lifecycle State specified by the method name.
testPrintArgs.py	testPrintArgs (*arg1*, *arg2*, ...)	Displays the specified argument values.

testPrintToFile.py	testPrintToFile (*arg1*)	Prints the specified argument values to a file named `actionFile` in the following directory:
		Windows:
		`<drive>:\ProgramData\HP\HP BTO Software\shared\nnm\actions`
		<*drive*> is the drive on which NNMi is installed.
		UNIX:
		`/var/opt/OV/shared/nnm/actions`

The output generated from these methods is written to the event action log. You can find the event action log in the following directory:

Windows:

`<drive>:\ProgramData\HP\HP BTO Software\log\nnm`

UNIX:

`/var/opt/OV/log/nnm/public`

Configure Diagnostics for an Incident (*NNM iSPI NET*)

HP Network Node Manager iSPI Network Engineering Toolset Software provides a set of Diagnostics (Flow Definitions) that can be run on the Source Node each time an incident reaches a specified Lifecycle State (for example, as soon as an incident becomes Registered).

Note: If you have the licensed HP Operations Orchestration (HP OO) product, you can import HP OO flow definitions into NNMi and then assign these flows to run when NNMi detects certain network incidents. See the "Custom HP OO Flow Management" section of the *HP NNM iSPI Network Engineering Toolset Software Planning and Installation Guide* and nnmooflow.ovpl for more information.

These Diagnostics are sets of automated commands specific to one or more device types, including Cisco routers and switches, Cisco switch/routers, and Nortel switches.

See "Configure Device Profiles" on page 292 for more information about device types . See "Diagnostics (Flows) Provided by NNM iSPI NET" below for more information about the Diagnostics provided by NNMi.

Configuring NNMi to automatically gather diagnostic information about the Source Node whenever a specified incident reaches a selected Lifecycle State is a two-step process:

1. Specify the Node Group providing the required information within one of the following contexts:
   - "Configure Node Settings for an SNMP Trap Incident" on page 840
   - "Configure Node Settings for a Syslog Message Incident (HP ArcSight)" on page 991
   - "Configure Node Settings for a Remote NNM 6.x/7.x Event Incident" on page 1278
   - "Configure Node Settings for a Management Event Incident" on page 1135
2. Specify the Diagnostics (Flow Definitions) providing the required information within one of the following contexts:
   - "Configure Diagnostics Selections for a Node Group (SNMP Trap Incident) (NNM iSPI NET)" on page 877
   - "Configure Diagnostics Selections for a Node Group (Syslog Message) (HP ArcSight)" on page 1027
   - "Configure Diagnostics Selections for a Node Group (Remote NNM 6.x/7.x Events)" on page 1315
   - "Configure Diagnostics Selections for a Node Group (Management Events)" on page 1171

Diagnostic Selections Form (*NNM iSPI NET*)

With HP Network Node Manager iSPI Network Engineering Toolset Software, the Diagnostic Selections form enables you to configure NNMi to automatically gather diagnostic information for the Incident you are configuring. When using this form, you specify the diagnostics you want to run on each applicable node in the specified Node Group.

You can provide the required information within the following contexts:

"Configure Diagnostics Selections for a Node Group (SNMP Trap Incident) (NNM iSPI NET)" on page 877

"Configure Diagnostics Selections for a Node Group (Remote NNM 6.x/7.x Events)" on page 1315

"Configure Diagnostics Selections for a Node Group (Management Events)" on page 1171

Note: If you have the licensed HP Operations Orchestration (HP OO) product, you can import HP OO flow definitions into NNMi and then assign these flows to run when NNMi detects certain network incidents. See the "Custom HP OO Flow Management" section of the *HP NNM iSPI Network Engineering Toolset Software Planning and Installation Guide* and nnmooflow.ovpl for more information.

Diagnostics (Flows) Provided by NNM iSPI NET

The HP Network Node Manager iSPI Network Engineering Toolset Software Diagnostics (Flows) are sets of automated commands specific to one or more device types. You can associate these

Diagnostics with specific incident configurations. After you associate a Diagnostic with an incident configuration and specify the Lifecycle State for which the Diagnostic should run, the Diagnostic automatically runs on the Source Node for the incident whenever the specified Lifecycle State is reached. See "Configure Diagnostics for an Incident (NNM iSPI NET)" on page 757 for more information.

Note: If you have the licensed HP Operations Orchestration (HP OO) product, you can import HP OO flow definitions into NNMi and then assign these flows to run when NNMi detects certain network incidents. See the "Custom HP OO Flow Management" section of the *HP NNM iSPI Network Engineering Toolset Software Planning and Installation Guide* and nnmooflow.ovpl for more information.

NNMi also associates these Diagnostics with each node to which the Diagnostics apply. To view the Diagnostics invoked for each node, open the Node form for any node of interest. See Node Form: Diagnostics Tab for more information.

NNMi provides Diagnostics (Flows) for the following device types:

- Cisco router
- Cisco switch
- Cisco switch/router (see Cisco router and Cisco switch)
- Nortel switch

Cisco Router Diagnostics (Flow Definitions) Provided by NNMi

Name	Description

Cisco Router Baseline Information	Uses a series of show commands to determine the current configuration of a Cisco router. It first displays the router's and NNMi management server's current times. Next, it invokes a series of commands on the router and formats these results on the summary page. Click here for a list of the commands included in this Diagnostic.
	`show version`
	`show protocol`
	`show interface summary`
	`show ip route`
	`show ip protocol`
	`show ip traffic`
	`show vlans`
	`show cdp`
	`show cdp entry`
	`show cdp neighbors`
	`show log`
	`show stacks`

Cisco Router Diagnostics (Flow Definitions) Provided by NNMi, continued

Name	Description
Cisco Show IP Route	Obtains routing information using the `show ip route` command.

Cisco Route To Node Diagnostic	**Note**: This Diagnostic Flow is not associated with an NNMi incident or node object and can only be run from Operations Orchestration Central's Flow Library. Before the Diagnostic runs, you are prompted for access information for the source router and target device node.
	Determines failures of either ping or traceroute to a target node. Uses the router to perform a ping and a traceroute to a target node.
	Click here for a list of commands included in this Diagnostic
	`ping target`
	`traceroute target`

Cisco Interface Diagnostic	Performs a number of diagnostic checks on a specified interface on the Cisco router. Diagnostics performed include whether the link is Down while the interface is Up. The following error counts are checked: • Input errors • CRC errors • Frame errors • Overrun errors • Ignored errors

Cisco Switch Diagnostics (Flow Definitions) Provided by NNMi

Name	Description

Cisco Switch Baseline Information	Uses a series of show commands to determine the current configuration of a Cisco switch. It first displays the switch's and NNMi management server's current times. Next, it invokes a series of commands on the switch and formats these results on the summary page.Click here for a list of the commands included in this Diagnostic.
	`show version`
	`show protocol`
	`show interface summary`
	`show vlans`
	`show cdp`

Cisco Switch Diagnostics (Flow Definitions) Provided by NNMi, continued

Name	Description

	`show cdp entry`
	`show cdp neighbors`
	`show log`
	`show stacks`

Cisco Switch Spanning Tree Baseline	Gathers spanning tree protocol and port information from the Cisco switch. The commands run depend on the device's operating system: IOS: show spanning-tree brief CATOS; show spantree

Nortel Switch Diagnostics (Flow Definitions) Provided by NNMi

Name	Description
Nortel Port Diagnostic	Determines statistics, including rate-limit and usage for a specified port on a Nortel switch. This Diagnostic detects rate limit, reception and transmission errors. Similar to Cisco Interface Diagnostic, this flow identifies the following types of errors on the identified port: • FCS errors • Undersized packets • Oversized packets • Collisions • Single collisions • Multiple collisions • Excessive collisions • Deferred packets • Late collisions

Nortel Route to Node Diagnostic	**Note**: This Diagnostic Flow is not associated with an NNMi incident or node object and can only be run from Operations Orchestration Central's Flow Library. Before the Diagnostic runs, you are prompted for access information for the source router and target device node.
	Determines failures of either ping or traceroute to a target node.
	Click here for a list of commands included in this Diagnostic
	`ping target`
	`traceroute target`

Nortel Switch Diagnostics (Flow Definitions) Provided by NNMi, continued

Name	Description

Nortel Switch Baseline	Determines the configuration of a Nortel switch. It first displays the switch's and NNMi management server's current times. Next, it invokes a series of commands on the switch and formats the results on the summary page. Click here for a list of commands included in this Diagnostic
	`show sys-info`
	`show interface`
	`show logging config`
	`show ssh global`
	`show stack-info`
	`send show rate-limit`
	`send show vlan`

Nortel Switch Spanning Tree Baseline	Gathers spanning tree protocol and port information from the Nortel switch. Click here for a list of commands included in this Diagnostic
	`show spanning-tree config`
	`show spanning-tree port`
	`show spanning-tree vlans`

Incident Configurations You Might Want to Enable

NNMi enables you to choose whether you want to generate an Incident for any Incident Configuration that is stored in the NNMi database. To do so you use the **Enable** attribute for each Incident Configuration.

Note: You can use the Actions menu from the NNMi console to Enable or Disable one or more Incident Configurations. See "Enable or Disable Configurations" on page 39 for more information.

By default, not all of the Incident Configurations NNMi provides are enabled.

To determine which Incident Configurations are enabled:

1. Navigate to the **Incidents** folder:
   a. From the workspace navigation pane, select the **Configuration** workspace.
   b. Expand the **Incidents** folder.
2. Select the incident configuration of interest (**SNMP Trap Configurations**, **Remote NNM 6.x/7.x Event Configurations**, or **Management Event Configurations**).
3. Click the **Enable** column heading to sort the incident configurations according to the **Enable** configuration setting.

   NNMi displays a ✓ check in the Enabled column for each incident configuration that is enabled.

You might want to enable the following incident configurations:

"Generate Interface Disabled Incidents" below

"Generate Card Disabled Incidents" below

"Generate Card Undetermined State Incidents" below

"Generate Performance Threshold Incidents (NNM iSPI Performance for Metrics)" on the next page

Generate Interface Disabled Incidents

By default, NNMi *does not generate* an incident for interfaces with **Administrative Status** set to **Down**. If you want NNMi to generate incidents for these disabled interfaces, use the following procedures.

To enable the Interface Disabled Management Event incident configuration:

1. Navigate to the **Incidents** folder.
   a. In the workspace navigation panel, select the 🔧 **Configuration** workspace.
   b. Expand the **Incidents** folder.
2. Select **Management Event Configurations**.
3. Double-click the row that represents the Interface Disabled configuration.
4. Click Enable ☑.

Generate Card Disabled Incidents

By default, NNMi *does not generate* an incident for cards with **Administrative Status** set to **Down**. If you want NNMi to generate incidents for these disabled cards, use the following procedures.

To enable the Card Disabled Management Event incident configuration:

1. Navigate to the Incidents folder.
   a. In the workspace navigation panel, select the 🔧 **Configuration** workspace.
   b. Expand the **Incidents** folder.
2. Select **Management Event Configurations**.
3. Double-click the row that represents the Card Disabled configuration.
4. Click Enable ☑.

Generate Card Undetermined State Incidents

By default, NNMi *does not generate* an incident for cards that have an undetermined State. (See Card Undetermined State for more information about these incidents.)

If you want NNMi to generate incidents for these cards, use the following procedures.

To enable the Card Undetermined State Management Event incident configuration:

1. Navigate to the **Incidents** folder.
   a. In the workspace navigation panel, select the 🔧 **Configuration** workspace.
   b. Expand the **Incidents** folder.
2. Select **Management Event Configurations**.
3. Double-click the row that represents the Card Undetermined State configuration.
4. Click Enable ☑.

Generate Node Deleted Incidents

By default, NNMi *does not generate* an incident for nodes that have been deleted from the NNMi topology.

If you want NNMi to generate incidents for these nodes, use the following procedures.

To enable the Node Deleted Management Event incident configuration:

1. Navigate to the **Incidents** folder.
   a. In the workspace navigation panel, select the 🔧 **Configuration** workspace.
   b. Expand the **Incidents** folder.
2. Select **Management Event Configurations**.
3. Double-click the row that represents the Node Deleted configuration.
4. Click Enable ☑.

Generate Performance Threshold Incidents (*NNM iSPI Performance for Metrics*)

NNMi can generate incidents related to performance thresholds. NNMi does not generate threshold incidents until the NNMi administrator configures the performance thresholds and enables the performance incidents.

To configure NNMi to generate performance threshold incidents:

1. *Prerequisite*. Enable performance polling and configure the performance thresholds. See "Configure Threshold Monitoring for Interface Groups" on page 381 for more information.
2. Navigate to the **Incidents** folder:
   a. From the workspace navigation pane, select the 🔧 **Configuration** workspace.
   b. Expand the **Incidents** folder.
3. Select **Management Event Configurations**.

> **Tip:** NNM iSPI Performance for Metrics threshold incidents are available in Management Event Configurations (not other categories of incident configurations).

4. Double-click the row representing the threshold incident configuration.

   For the list of possible values, see Management Event Configurations Provided by NNMi.

5. Enable the threshold incident:

   a. **Management Event Configuration** form

   b. **Basics** group

   c. Select **Enable**☑

6. Click **Save and Close** to save your changes.

7. Repeat steps 4 through 7 for each configuration you want to use.

The NNM iSPI Performance for Metrics now records the number and frequency of threshold related incidents (exceptions). The NNM iSPI Performance for Metrics provides reports to help you establish the root cause of network problems. Access the NNM iSPI Performance for Metrics reports with **Actions** → **HP NNM iSPI Performance** → **Reporting - Report Menu** in the incident, node, or interface views and forms. (See NNM NNM iSPI Performance for Metrics Actions.)

Using the Command Line to Manage Incident Configurations

You can use the nnmincidentcfgload.ovpl script to generate a file of your Incident Configurations and then load them into the NNMi database.

Incident Configurations are exported in a non-xml format. You can edit the file using the format descriptions provided in nnmincidentcftg.format and in the following directory:

Windows

`%NnmInstallDir%/examples/nnm/incidentcfg`

UNIX

`/opt/OV/examples/nnm/incidentcfg`

"Generate a File of Your Incident Configurations" below

"Load Incident Configurations Using the Command Line" on page 768

Generate a File of Your Incident Configurations

Tip: If you do not want to enter an NNMi User Name attribute value and an NNMi Password attribute value at the command line, you can use the `nnmsetcmduserpw.ovpl` command to specify the valid user name and password (instead of `-u` and `-p`). The credentials set using the `nnmsetcmduserpw.ovpl` command are valid for command execution by the same user. See "Set Up Command Line Access to NNMi" on page 577 for more information.

The NNMi `nnmincidentcfgdump.ovpl` script provides a way for you to create or update an Incident Configuration to subsequently load into the NNMi database using the `nnmincidentcfgload.ovpl` script. The file is generated in a non-xml format.

You can edit the file using the format descriptions provided in the following directory:

Windows

```
%NnmInstallDir%/examples/nnm/incidentcfg
```

UNIX

```
/opt/OV/examples/nnm/incidentcfg
```

To generate a file of your Incident Configurations, use the following example syntax:

```
nnmincidentcfgdump.ovpl -dump <file_name> -u <NNMiadminUsername> -p
<NNMiadminPassword>
```

Note: See `nnmincidentcfgdump.ovpl` for more information, including a complete list of the valid script arguments.

nnmincidentcfg.ovpl Arguments

Argument	Description

-dump <*file_ name*>	Used to create a file of your Incident Configurations.
	Tip:Incident Configurations can be loaded into the NNMi database using the `nnmincidentcfgload.ovpl` script.
	To create an Incident Configuration file without using existing Incident Configurations, start with one of the template files provided in the following directory:
	Windows
	`%NnmInstallDir%/examples/nnm/incidentcfg`
	UNIX
	`/opt/OV/examples/nnm/incidentcfg`

-uuid	*Recommended*. Specifies the Universally Unique Object Identifier (UUID) for each configuration entry. **Note**: Configuration files that do not contain the UUID value take longer to load. See `nnmincidentcfgload.ovpl` for more information.
-authorKey <*author or authors*>	*Optional*. Generates an Incident Configuration file that contains the settings created by one or more authors. Note the following: • You can include one or more <*author*> values. • If you do not specify any configuration authors, NNMi includes all of the incident configurations. • You cannot use this argument with the `-name` argument. **To find a Unique Key for a particular author using the command line, execute**: `nnmicidentcfgdump.ovpl -ListAuthors`

nnmincidentcfg.ovpl Arguments, continued

Argument	Description
	To find the Unique Key for a particular Author in the NNMi console: 1. Open one of the Incident Configuration workspace in the NNMi console. 2. Select an object created by the Author of interest. 3. Display the Author form, and copy the value of the Unique Key attribute.
-u	*Optional*. The NNMi user name. This User Account must be assigned to the **NNMi Administrators** User Group.
-p	*Optional*. The password associated with the NNMi user name.
-name <*name or names*>	*Optional*. Specifies the name of each configuration that should be included in the configuration file you are creating. Note the following: • If you do not specify any configuration Names, NNM includes all of the configurations. • You cannot use this argument with the `-authorKey` argument.
-type <*config_ type*>	*Optional*. Specifies the type of configurations you want to include. Valid configuration types include: • MgmtEventConfig • PairwiseConfig • RemoteNNmEventConfig • SnmpTrapConfig • SyslogMessageConfig
-mib <*module_ name*> or <*module_ names*>	Specifies the MIB module or modules that must be contained in an incident configuration to be included in the formatted configuration file. NNMi includes any SNMP Trap Configurations that contain the specified MIB module. **Tip**: MIB modules are loaded from MIB files using the nnmloadmib.ovpl script. To see what MIB modules are loaded, use the nnmloadmib.ovpl script with the -list option.
-oid <*oid_ pattern* or *oid_ patterns*>	Specifies the Object Identifier (OID) pattern or patterns that must be contained in an incident configuration to be included in the formatted configuration file.NNMi includes any SNMP Trap Configurations that match a specified OID pattern. Each OID pattern can contain one wildcard character (*). See `nnmincidentcfgload.ovpl`for more information.

Load Incident Configurations Using the Command Line

Tip: If you do not want to enter an NNMi User Name attribute value and an NNMi Password attribute value at the command line, you can use the nnmsetcmduserpw.ovpl command to specify the valid user name and password (instead of `-u` and `-p`). The credentials set using the `nnmsetcmduserpw.ovpl` command are valid for command execution by the same user. See "Set Up Command Line Access to NNMi" on page 577 for more information.

The NNMi `nnmincidentcfgload.ovpl` script provides a way for you to load Incident Configurations into the NNMi database from a formatted configuration file.

Tip: Use the `nnmincidentcfgdump.ovpl` script to create a configuration file of existing Incident Configurations in a non-xml format. You can then edit this file if desired before loading them into the NNMi database.

See the following directory for the required format:

Windows

```
%NnmInstallDir%/examples/nnm/incidentcfg
```

UNIX

```
/opt/OV/examples/nnm/incidentcfg
```

To validate an Incident Configuration file before it is loaded into the NNMi database, use the following example syntax:

```
nnmincidentcfgload.ovpl -validate <file_name> -u <NNMiadminUsername> -
p <NNMiadminPassword>
```

To load Incident Configurations, use the following example syntax:

```
nnmincidentcfgload.ovpl -load <file_name> -u <NNMiadminUsername> -p
<NNMiadminPassword>
```

Note the following:

- NNMi updates all configurations that have matching names or other matching key identifiers.

  Caution: NNMi also overwrites the values of any codes associated with these configurations (for example, incident Family).

- NNMi adds all incident configurations with key identifiers that do not exist in the NNMi database.
- NNMi does not change existing incident configurations with key identifiers that do not match any in the exported file.
- NNMi resolves Universally Unique Object Identifiers (UUIDs) if they are not provided in the configuration file.
- If NNMi is unable to resolve a UUID, a UUID is created.

See `nnmincidentcfgload.ovpl` for more information, including a complete list of the valid script arguments.

nnmincidentcfg.ovpl Arguments

Argument	Description

-load <*file_ name*>	Use to load the Incident Configurations generated using either the `nnmincidentcfgdump.ovpl` script or created from a template file provided by NNMi.
	To create an Incident Configuration file without using existing Incident Configurations, start with one of the template files and required formats provided in the following directory:
	Windows
	`%NnmInstallDir%/examples/nnm/incidentcfg`
	UNIX
	`/opt/OV/examples/nnm/incidentcfg`

-validate <*file_ name*>	*Optional.* Use to validate the contents of the Incident Configuration file generated using nnmincidentcfgdump.ovpl.
-expression < *expression* >	Specifies the filter expression you want to validate.
-u	*Optional*. The NNMi user name. This User Account must be assigned to the **NNMi Administrators** User Group.
-p	*Optional*. The password associated with the NNMi user name.

Manage Incoming SNMP Traps

NNMi provides several tools that enable you to manage the SNMP traps that are sent through the Event Pipeline and are configured to appear as incidents in the NNMi console. For more information about NNMI's Event Pipeline, see "About the Event Pipeline" on page 603.

NNMi uses the following criteria to determine whether it *receives or discards incoming* traps:

- If the *incoming* trap's Source Node object or Source Object (such as card or interface) has not yet been discovered, NNMi discards the trap by default.

  Note: The NNMi administrator can change this behavior using the **Trap Handling Settings** when configuring incidents. See "Handle Unresolved Incoming Traps" on page 776 for additional information. See also "Configure Network Devices to Send SNMP Notifications to NNMi" on page 771.

- If the Source Node or Source Object of the *incoming* trap has been discovered by NNMi using SNMPv3, NNMi accepts *incoming* traps from SNMPv3, SNMPv2c, or SNMPv1. See SNMPv3 Settings Form for information about configuring SNMPv3 settings.

- If the Source Node or Source Object of the *incoming* trap has been discovered by NNMi using SNMPv2c or SNMPv1, NNMi discards *incoming* traps from SNMPv3.
- NNMi discards traps that have no incident configuration or with an incident configuration set to Disabled. To ensure that NNMi retains all received Trap instances when your network environment includes SNMP agents using a variety of SNMPv1, SNMPv2c, and SNMPv3 protocol, you must configure two Incidents: one for the SNMPv1 version and one for the SNMPv2c/3 version of that trap. See "Configure SNMP Trap Incidents" on page 782.
- If either the Source Node or Source Object has *Management Mode* set to **Not Managed** or **Out of Service** in the NNMi database, NNMi always discards the incoming trap. See "Understand the Effects of Setting the Management Mode to Not Managed or Out of Service " on page 464.

  NNMi provides the Management Mode workspace so that you can quickly view lists of all nodes, interfaces, cards, addresses, or node components that NNMi is not currently discovering or monitoring. For information about these views:
- NNMi discards most incoming traps from network objects that are not monitored. For example, you can configure NNMi to exclude specified interfaces from being monitored. See "Monitoring Network Health" on page 340 for more information.

Note the following:

- If you want the NNMi management server to *forward* traps to other machines in your network environment, see the following topics for additional information and configuration steps:
  - SNMPv2c traps — "Configure Trap Forwarding" on page 1376
  - SNMPv3 traps — "Configure NNMi SNMPv3 Security Settings for Trap Forwarding and Inform-Requests" on page 1376
- NNMi administrators can configure thresholds for trap volume within your network. Choices include count-based or time-based thresholds for total volume or for each trap OID. Traps can also be blocked. See the following Reference Pages:
  - nnmtrapconfig.ovpl
  - hosted-object-trapstorm.conf

When managing your SNMP Traps consider performing the following tasks:

"Configure Network Devices to Send SNMP Notifications to NNMi" on the next page

"Load SNMP Trap Incident Configurations" on the next page

"Control which Incoming Traps Are Visible in Incident Views" on page 775

"Handle Unresolved Incoming Traps" on page 776

"Analyze Trap Information" on page 776

Related Topics:

"Configure Trap Forwarding" on page 1376

Configure Network Devices to Send SNMP Notifications to NNMi

An SNMP notification is a message sent from an SNMP agent (SNMPv1, SNMPv2c, or SNMPv3) on a network device to notify a network management system of an event on the network device. For example, an error occurred on the network device and its SNMP agent sent a notification. The notification might be either of the following:

- An acknowledged inform (SNMP InformRequest): An inform is an acknowledged notification sent from one SNMP agent to another with the expectation of a reply from the recipient. If no reply is received, the inform message is resent.
- An unacknowledged trap: A trap is a notification sent from one SNMP agent to another without any expectation of a reply.

Configure SNMP agents in your network environment to send traps to the NNMi management server. Sometimes SNMP agents are configured with a recheck interval, so the trap might be sent to the NNMi management server over and over again until the problem is corrected.

The NNMi Causal Engine analyzes these traps and gathers additional information to determine the root cause. It also provides useful troubleshooting information each time an important SNMP notification is received, including the following information:

- The name or address of the node from which the notification came (Source Node)
- The notification identification (SNMP Object ID)
- Notification-specific variables (varbinds)

When configuring the SNMP agent for each network device, configure the trap-forwarding list (or trap-destination list) to include the NNMi management server's fully-qualified hostname or IP address. Refer to documentation for the SNMP agent for information about how to do this. If the NNMi management server is included on the trap-forwarding list, NNMi receives notice when something goes wrong (even if the device does not show up on your NNMi maps).

Note: For an SNMP notification to be processed by NNMi, it must be configured using the NNMi Incidents folder workspace. Many common SNMP notifications are configured in NNMi by default. See "Configure SNMP Trap Incidents" on page 782 and "SNMP Trap Incident Configurations Provided by NNMi" on page 612 for more information.

Load SNMP Trap Incident Configurations

NNMi enables you to automatically create or update an Incident Configuration for an SNMP trap using a MIB file. To load a trap definition using a MIB file, you can use either the command line or NNMi console:

"Load SNMP Trap Incident Configurations from the Command Line" on the next page

"Load SNMP Trap Incident Configurations using the Console" on page 774

When loading MIBs to be used for SNMP Trap Incident configurations, NNMi stores TRAP-TYPE or NOTIFICATION-TYPE Macro Definition information from the MIB into the NNMi database.The following table lists field names included in the Macro Definition information.

> Note: If any of the individual field lengths in a TRAP-TYPE or NOTIFICATION-TYPE Macro Definition for a MIB exceeds the limitations for storing this information in the NNMi database, the MIB will not load into NNMi. The most commonly encountered field limitations are listed in the following table.

TRAP-TYPE or NOTIFICATION-TYPE Macro Definition Maximum Field Lengths

Field Name	Maximum Length	Multi-Entry
NAME	80	NO
DESCRIPTION	4000	NO
--#TYPE	255	NO
--#SUMMARY	2000	YES
--#ARGUMENTS	255	NO
--#SEVERITY	255	NO
--#GENERIC	40	NO
--#CATEGORY	80	NO
--#SOURCE_ID	40	NO
--#TIMEINDEX	40	NO
--#HELP	80	NO
--#HELPTAG	40	NO
--#STATE	80	NO

Load SNMP Trap Incident Configurations from the Command Line

Tip: If you do not want to enter an NNMi User Name attribute value and an NNMi Password attribute value at the command line, you can use the nnmsetcmduserpw.ovpl command to specify the valid user name and password (instead of -u and -p). The credentials set using the nnmsetcmduserpw.ovpl command are valid for command execution by the same user. See "Set Up Command Line Access to NNMi" on page 577 for more information.

The NNMi nnmincidentcfg.ovpl script provides a way for you to create or update an Incident Configuration for an SNMP trap using a MIB module that was previously loaded into the NNMi database using the nnmloadmib.ovpl script with the -load option. To load a MIB module you can use the following syntax:

```
nnmincidentcfg.ovpl -loadTraps <mib_module_name> -disableAllTraps
true|false -u <NNMiadminUsername> -p <NNMiadminPassword>
```

Note: See nnmincidentcfg.ovpl for more information, including a complete list of the valid script arguments.

nnmincidentcfg.ovpl Arguments

Argument	Description
-loadTraps <*mib_module_name*>	Used to load the traps from the specified MIB module you want to use to create or update the incident configuration for an SNMP trap. **Tip**:MIB modules are loaded from MIB files using the nnmloadmib.ovpl script. To see what MIB modules are loaded, use the nnmloadmib.ovpl script with the `-list` option. NNMi uses information from the trap definitions (TRAP-TYPES macro) or notification (NOTIFICATION-TYPES macro) in the MIB module for the required incident configuration.
-disableAllTraps	Specifies whether all trap definitions specified using `-loadTraps` <*mib_module_name*> should be loaded as disabled. **Note**: The default value is `false`. This means that by default all trap definitions specified in <*mib_module_name*> are loaded as enabled. Set this parameter to `true` to disable the trap definitions that you are loading.
-u	The NNMi user name. This User Account must be assigned to the **NNMi Administrators** User Group. **Note:** The user name might be a Principal object stored in the NNMi database or might be from Lightweight Directory Access Protocol (LDAP) or X.509 Certificates such as Public Key Infrastructure (PKI) user authentication in your environment. See "Choose a Mode for NNMi Access" on page 504.
-p	The password associated with the NNMi account. If you do not want to enter an NNMi User Name attribute value and an NNMi Password attribute value at the command line, you can use the nnmsetcmduserpw.ovpl command to specify the valid user name and password (instead of `-u` and `-p`). The credentials set using the `nnmsetcmduserpw.ovpl` command are valid for command execution by the same user. See "Set Up Command Line Access to NNMi" on page 577 for more information.

For example, to load the MIB module CISCO-VTP-MIB, you might enter the following:

```
nnmincidentcfg.ovpl -loadTraps "CISCO-VTP-MIB"
```

If the incident is already configured, NNMi performs an update based on the MIB file information. If the incident is not configured, NNMi creates a new incident configuration entry for the SNMP trap. See "Configure SNMP Trap Incidents" on page 782 for information about changing an SNMP trap configuration.

Load SNMP Trap Incident Configurations using the Console

NNMi enables you to load one or more SNMP Trap Incident Configurations from a MIB file using the NNMi console.

To load an SNMP Trap Incident Configuration from the NNMi console:

1. Do one of the following:
   a. Navigate to the MIB view or form. For example, Select **Configuration** → **MIBs** → **Loaded MIBs**.

   b. Navigate to the MIB Variable view or form. For example, Select **Inventory** → **MIB Variables**.
2. Select **Tools** → **Load/Unload MIB...**

   NNMi displays the following information:

   - Unloaded MIBs (user provided) that are stored on the NNMi management server and that were provided by the NNMi administrator.
   - Unloaded MIBs (NNMi provided) that NNMi has stored on the NNMi management server during installation.
   - Loaded MIBs that are loaded in the NNMi database.
3. Navigate to the Unloaded MIB view of interest. For example, **Unloaded MIBs (User Provided)**.
4. In the MIB column, find the MIB that contains the trap incidents you want to load. For example, **FLOWMGREST-MIB**.

   Note: The MIB must support the TRAP-TYPE or NOTIFICATION-TYPE macro.
5. To view the MIB before loading, in the Actions column, click **Display**.

   NNMi displays the MIB file contents.
6. To load the MIB, in the Actions column, click **Load Incident Configuration**.

   NNMi displays the progress of each trap configuration that is loaded, including the following:

   - The name and location of the MIB file
   - The number of trap incident configurations
   - The name and numeric object identification (OID) of each trap configuration
   - Whether the trap incident configurations successfully loaded
   - Instructions for loading and listing MIB files.

To upload a local MIB file so that it is stored on the NNMi management server and available for loading, see "Upload MIB Files from the Console" on page 1457.

Control which Incoming Traps Are Visible in Incident Views

You can configure devices in your network environment to send traps to the NNMi management server. To configure how NNMi handles those traps, use the incident configurations provided by NNMi, create your own, or both. See "Configure SNMP Trap Incidents" on page 782 for information about how to configure SNMP traps as incidents. See "SNMP Trap Incident Configurations Provided by NNMi" on page 612 for information about the incident configurations provided by NNMi.

Note: To establish this communication flow, the SNMP agent (SNMPv1, SNMPv2c, or SNMPv3) must be intentionally configured by the device administrator to send SNMP traps to your NNMi management server.

After you configure an incident for each SNMP trap you want NNMi to display, NNMi stores your incident configurations for SNMP traps in the `allowedOids.conf` file. NNMi uses this file as a positive filter to identify which traps should appear as incidents.

To determine which SNMP Trap incident configurations are Enabled in NNMi:

1. Navigate to the **SNMP Trap Incident Configurations** view:
   a. Expand the **Incidents** folder.

   b. Select **SNMP Trap Incident Configurations**.

2. Click to sort the **Enabled** column.

   Each SNMP Trap Incident Configuration that is Enabled contains a check mark ✓.

See "SNMP Trap Incident Configurations Provided by NNMi" on page 612 for more information.

Tip: You can configure NNMi to ignore SNMP Traps for objects that are not discovered as part of the NNMi topology. See "Handle Unresolved Incoming Traps" on the next page for more information.

To enable or disable an SNMP trap configuration:

1. Navigate to the Incidents folder.

   a. In the Workspace navigation panel, select the **Configuration** workspace.

   b. Expand the **Incidents** folder.

2. Select **SNMP Trap Configurations** .

3. Double-click the row representing the configuration you want to edit.

4. To enable the incident configuration, click Enable ☑.

5. To disable the incident configuration, clear Enable □.

Related Topics

"Handle Unresolved Incoming Traps" on the next page

"Manage the Number of Incoming Incidents" on page 653

Handle Unresolved Incoming Traps

Your network environment might be configured to forward SNMP traps to the NNMi management server.

If the trap's source node or source object *cannot be matched with any object in the NNMi database*, that trap is considered to be *unresolved*. Follow the steps in this procedure to specify whether NNMi retains or discards these traps. For example, if you configure NNMi to discover only devices you specifically list as seeds, you can decide if you want NNMi to process or ignore traps from any other devices.

See "Manage Incoming SNMP Traps" on page 769 for more information about the additional criteria NNMi uses to determine when to receive or discard traps.

To control how NNMi handles unresolved incoming SNMP Traps:

1. Navigate to the **Incident Configuration** form:
   a. From the workspace navigation pane, select the **Configuration** workspace.
   b. Expand the **Incidents** folder.
   c. Select **Incident Configuration...**.
2. Navigate to the **NNMi Trap Handling Settings**:
   - If you want NNMi to place unresolved SNMP traps into the NNMi database, clear the **Discard Unresolved SNMP Traps** □ check box.

     Unresolved traps then appear in incident views, but have missing information. For example, the incident might appear as follows:

     - NNMi displays the Source Node as an IP address.
     - NNMi displays the Source Object as **None**.
   - If you want NNMi to ignore any unresolved traps, select the **Discard Unresolved SNMP Traps** ☑ check box.
3. Select **Save and Close** to save your changes.

Tip: To manage the number of SNMP Traps displayed as incidents, see "Control which Incoming Traps Are Visible in Incident Views" on the previous page

Analyze Trap Information

NNMi measures the rate of incoming SNMP traps regardless of Incident Configuration, including the following:

- Traps from each Node.
- Traps for each SNMP Object Identifier (OID).

NNMi monitors the incoming SNMP traffic flow to determine whether the number of traps received within a certain time period exceeds any set threshold. If a threshold is exceeded, NNMi blocks processing of additional traps until the number of traps is below the threshold set for each time period.

> **Note:** The NNMi administrator configures thresholds for trap volume within your network. Choices include count-based or time-based thresholds for total volume or for each trap OID. You can also block traps. See the following Reference Pages:
>
> - nnmtrapconfig.ovpl
> - hosted-object-trapstorm.conf

When analyzing traps, NNMi looks at both the most common traps as well as the most common Source Nodes from which the traps are received. NNMi logs this SNMP trap analytics data to the `trapanalytics.0.0.log` file.

If NNM iSPI NET is available in your network environment, you can obtain reports about incoming SNMP traps according to the criteria described in the Trap Analytics Reports table.

Note the following:

- The time interval and number of Nodes or SNMP OIDs included in the reports and Line Graphs is based on the numbers configured using the nnmtrapconfig.ovpl script. By default, NNMi uses 5 minutes as the time interval and 10 as the top number of Nodes and SNMP OIDs for which information is computed.
- NNMI identifies each trap using its SNMP Object Identifier (OID) number.
- NNMi enables you to open the following graphs, reports, and forms from a Trap Analytics report:
  - Line Graph of the trap rate for all of the Nodes or SNMP OIDs displayed in the report.
  - Line Graph of the trap rate for a selected Node or SNMP OID.
  - SNMP Trap Incident Configuration form, if any, for an SNMP OID.
  - Source IP Address and Node form for a Node.

    Note: The Source Node must be stored in the NNMi database for the links to appear.
  - MIB Variable form, if any, for the selected SNMP OID.

    Note: The MIB Variable must be stored in the NNMi database. To add a MIB Variable by loading a trap, see "Load SNMP Trap Incident Configurations" on page 771
- When you access a Line Graph from a report, the Line Graph displays an updated real-time data using the Nodes or SNMP OIDs included in the report. Because the trap rate is constantly changing, the data in the Line Graph will not match the historical trap numbers displayed in the report.
- If an SNMP Trap Incident Configuration exists for a trap, NNMi displays the name of the SNMP Trap Incident Configuration as well as whether the SNMP Trap Incident Configuration is disabled. This feature is useful when you want to make changes to the incident configuration. For example, you might want to enable or disable the incident configuration.
- NNMi discards traps that have no incident configuration or with an incident configuration set to Disabled. To ensure that NNMi retains all received Trap instances when your network environment includes SNMP agents using a variety of SNMPv1, SNMPv2c, and SNMPv3 protocol, you must configure two Incidents: one for the SNMPv1 version and one for the SNMPv2c/3 version of that trap. See "Configure SNMP Trap Incidents" on page 782.

See the Trap Analytics Reports table for more information about the links available from each report.

To access the Trap Analytics reports:

1. Select **Tools** → **Trap Analytics (iSPI NET only)**.
2. Select the graph or report of interest. from the **Trap Analytics** submenu.

   NNMi displays the selected report (see Trap Analytics Reports).

Trap Analytics Reports

Report	Description	Links Available from the Report
Recent Top Trap Rate (by Node)	Table view of the Nodes that are most frequently generating traps during the specified time period.	Line Graph of the Nodes that are most frequently generating traps. Recent Top Rate Traps Received (by OID) report Total Traps Received (by Node) Total Traps Received (by OID) Line Graph of the trap rate for the selected Node. Source Node form, if any, for the trap.
Recent Top Trap Rate (by OID)	Table view of the traps that are most frequently generated during the specified time period.	Line Graph of the traps that are most frequently generated. Recent Top Rate Traps Received (by Node) report Total Traps Received (by Node) Total Traps Received (by OID)

Trap Analytics Reports, continued

Report	Description	Links Available from the Report
		Line Graph of the trap rate for the selected SNMP OID. Incident Configuration form, if any, for the selected SNMP OID. MIB variable form, if any, for the MIB variable that is associated with the SNMP OID.
Total Traps Received (by Node)	Table view of the trap totals since NNMi was last started. This report is organized by traps per Node.	Line Graph of the total number of traps received per Node since NNMi was last started. Total Traps Received (by OID) report Recent Top Trap Rate (by Node) Recent Top Trap Rate (by OID) Line Graph of a selected Node's total traps in real time. Source Node form, if any, for the selected trap.
Total Traps Received (by OID)	Table view of the trap totals since NNMi was last started. This report is organized by traps per SNMP OIDs.	Line Graph of the total number of traps received since NNMi was last started.

Trap Analytics Reports, continued

Report	Description	Links Available from the Report
		Total Traps Received (by Node) report Recent Top Trap Rate (by Node) Recent Top Trap Rate (by OID) Line Graph of the selected SNMP OID's total traps in real time. Incident Configuration form, if any, for the selected SNMP OID. MIB variable form, if any, for the MIB variable that is associated with the SNMP OID.
Trap Analysis Log	Log file listing trap information organized by the following criteria: • Trap rate in number of traps per second • The top 10 addresses that are generating traps • The top 10 traps that are being generated This information is recomputed every 5 minutes as configured in nnmtrapconfig.ovpl. Scroll to the bottom to see the latest entry. **Note**: The NNMi administrator can configure threshold values using the nnmtrapconfig.ovpl script. You can also use the `nnmtrapdump.ovpl` command to extract the data in which you are most interested from the `trapanalytics.0.0.log` file. See the nnmtrapdump.ovpl Reference Page for more information (**Help** → **Documentation Library** → **Reference Pages**, in the *User Commands* category).	

Control the Times within which NNMi Causal Engine Accepts SNMP Traps

When large areas of a network are unavailable at regular and predictable hours, NNMi enables you to moderate Causal Engine analysis load by inhibiting the delivery of traps to the Causal Engine. To inhibit the delivery of traps, as an NNMi administrator, you configure times that the NNMi Causal Engine stops accepting traps from the event system.

> **Note:** This feature does not interfere with traps delivered to the NNMi console.

Traps that are delivered to the Causal Engine are used to trigger State Poller to poll a node sooner than the schedule dictated by the State Poller Polling Policy. When you inhibit the delivery of traps, NNMi must wait until the scheduled polling interval before obtaining updated information from State Poller. In all cases, the NNMi Causal Engine reaches the same conclusion with or without traps by using state flows from the NNMi State Poller.

See "Maintaining NNMi" in the HP Network Node Manager i Software Deployment Reference for more information.

Configure Incident Logging

NNMi enables you to configure incident logging so that incoming incident information is written to the `incident.csv` file. This feature is useful when you want to track and archive your incident history.

Tip: You can also use the nnmtrimincidents.ovpl command to configure incident logging.

The `incident.csv` is located in the following directories:

Windows

`%NnmDataDir%\log\nnm`

UNIX

`$NnmDataDir/log/nnm`

To configure incident logging:

1. Navigate to the **Incidents** folder.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Expand the **Incidents** folder.
2. Select **Incident Configuration**.
3. Navigate to the **Incident Logging Configuration** tab.
4. Provide the required information (see General Configuration and Log File Configuration).
5. Click **Save and Close** to save your changes.

If **Enable Incident Logging**☑, the next time an incident arrives, NNMi logs the information to the `incident.csv` file.

Note: See nnmtrimincidents.ovpl for a description of the incident information that is written to the `incident.csv` file.

General Configuration

Attribute	Description
Enable Incident Logging	If enabled ☑, NNMi logs incoming incident information to the `the incident.csv` file. If disabled □, incident information is not logged.

Log File Configuration

Attribute	Description
Enable Compression	If enabled ☑, NNMi saves the `incident.csv` file in compressed (.gz) format. If disabled □ , incident information is not saved in the compressed (.gz) format.
Maximum File Size (MB)	Specify the maximum amount of disk space in megabyte that NNMi should use for the `incident.csv` file. The default value is 128 megabytes. **Note**: After the maximum file size is reached, the log file is renamed to `incident.csv.<gz>.old` and a new `incident.csv` file is created. If an `incident.csv.<gz>.old` exists, it is overwritten.
Logging Interval (ms)	Specify the time interval in which NNMi should log incident information. The default value is 6 seconds (6000 milliseconds). **Tip**: To optimize performance, use a less frequent Logging Interval with a larger Maximum Number of Incidents. Note the following: • The minimum value is 1 second (10 millieseconds). • The maximum value is 1 minute (60000 milliseconds).
Maximum Number of Incidents per Logging Interval	Specify the number of incidents to be logged to the `incident.csv` file per the **Logging Interval** specified. The default value is 1024. **Tip**: To optimize performance, use a less frequent Logging Interval with a larger Maximum Number of Incidents. The minimum value is 32.

Configure SNMP Trap Incidents

Configure incidents that originate from an SNMP trap.

Create one Trap Incident configuration for each trap (separate configurations for an SNMPv2 trap number and a similar SNMPv1 trap number). For example:

- .1.3.6.1.4.1.11.15.1.4.1 (SiteScopeAlertEventv2)
- .1.3.6.1.4.1.11.15.1.4.0.1 (SiteScopeAlertEventv1)

NNMi discards traps that have no Incident Configuration or with an Incident Configuration set to Disabled. To ensure that NNMi retains all received Trap instances when your network environment includes SNMP agents using a variety of SNMPv1, SNMPv2c, and SNMPv3 protocol, you must configure two Incidents: one for the SNMPv1 version and one for the SNMPv2c/3 version of that trap.

Tip: You can manage the number of SNMP Traps using either of the following methods: 1) "Manage the Number of Incoming Incidents" on page 653 and 2) "Handle Unresolved Incoming Traps" on page 776.

Note: See "Manage Incoming SNMP Traps" on page 769 for information about the criteria NNMi uses to determine when to receive or discard traps.

To configure incidents originating from SNMP traps:

1. Navigate to the **Incidents** folder.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Expand the **Incidents** folder.
2. Select **SNMP Trap Configurations** .
3. Do one of the following:
   - To create an SNMP trap configuration, click the ✱ New icon, and continue.
   - To edit an SNMP trap configuration, double-click the row representing the configuration you want to edit, and continue.
   - To delete an SNMP trap configuration, select a row, click the ✖ Delete icon.
4. In the SNMP Traps form, provide the required information.
5. Click **Save and Close** to save your changes.

The next time that a trap of this type arrives, NNMi creates an associated incident to display in the appropriate incident views.

SNMP Trap Configuration Form

Create one Trap Incident configuration for each trap (separate configurations for an SNMPv2 trap number and a similar SNMPv1 trap number). For example:

- .1.3.6.1.4.1.11.15.1.4.1 (`SiteScopeAlertEventv2`)
- .1.3.6.1.4.1.11.15.1.4.0.1 (`SiteScopeAlertEventv1`)

Note: See "Manage Incoming SNMP Traps" on page 769 for information about the criteria NNMi uses to determine when to receive or discard traps.

To configure incidents originating from SNMP traps:

1. Navigate to the **Incidents** folder:
   a. From the workspace navigation pane, select the **Configuration** workspace.
   b. Expand the **Incidents** folder.
2. Select **SNMP Trap Configurations** .
3. Do one of the following:

   > **Note:** If you want to add or edit an SNMP trap configuration, verify that **Enabled** ☑ is selected.

   - To add an SNMP trap configuration, click the ✳ New icon, and continue.
   - To edit an SNMP trap configuration, double-click the row representing the configuration you want to edit, and continue.
   - To delete an SNMP trap configuration, select a row, and click the ✖ Delete icon.
4. Make your configuration choices (see table).
5. Click **Save and Close** to save your changes and return to the previous form.

Tasks for SNMP Trap Configuration

Task	How
"Configure Basic Settings for an SNMP Trap Incident" on the next page	Use the **Basics** pane of the SNMP Trap Configuration form.
"Configure Interface Settings for an SNMP Trap Incident" on page 803	Use the **Interface Settings** tab of the SNMP Trap Configuration form.
"Configure Node Settings for an SNMP Trap Incident" on page 840	Use the **Node Settings** tab of the SNMP Trap Configuration form.
"Configure Suppression Settings for an SNMP Trap Incident" on page 879	Use the **Suppression** tab of the SNMP Trap Configuration form.
"Configure Enrichment Settings for an SNMP Trap Incident" on page 887	Use the **Enrichment** tab of the SNMP Trap Configuration form.
"Configure Dampening Settings for an SNMP Trap Incident" on page 892	Use the **Dampen** tab of the SNMP Trap Configuration form.
"Configure Deduplication for an SNMP Trap Incident" on page 901	Use the **Deduplication** tab of the SNMP Trap Configuration form.
"Configure Rate (Time Period and Count) for an SNMP Trap Incident" on page 909	Use the **Rate** tab of the SNMP Trap Configuration form.
"Configure Actions for an SNMP Trap Incident" on page 913	Use the **Actions** tab of the SNMP Trap Configuration form.
"Configure Forward to Global Manager Settings for an SNMP Trap Incident (NNMi Advanced)" on page 927	Use the **Forward to Global Managers** tab of the SNMP Trap Configuration form.

Configure Basic Settings for an SNMP Trap Incident

The Basics settings for an SNMP Trap Incident specifies general information for an incident configuration, including the name, severity, and message.

Note the following:

- NNMi discards traps that have no Incident Configuration or with an Incident Configuration set to Disabled. To ensure that NNMi retains all received Trap instances when your network environment includes SNMP agents using a variety of SNMPv1, SNMPv2c, and SNMPv3 protocol, you must configure two Incidents: one for the SNMPv1 version and one for the SNMPv2c/3 version of that trap.
- When configuring SNMP Trap incidents, if you are using SNMPv3 protocol:
    - You must also configure SNMPv3 communication using the Communication Configuration workspace. For more information,
    - If you configured SNMPv3 communication, use the **Actions** → **Configuration Settings** → **Communication Settings** to determine the SNMPv3 user name that NNMi will use for any node from which you want to receive SNMP Trap incidents. Make sure the node is configured with this user name when configuring SNMP trap incidents. See "Troubleshooting Communication Settings" on page 169 for more information.
    - If you configured SNMPv1 or SNMPv2c communication, NNMi does not authenticate the community string for any node from which you want to receive SNMP Trap incidents.
- In the **Basics** group of the **SNMP Trap Configuration** form, verify that **Enable** ☑ is selected for each configuration you want to use.

For information about each SNMP Traps tab:

To configure Basic settings for an SNMP Trap incident:

1. Navigate to the **SNMP Trap Configuration** form:
    a. From the workspace navigation panel, select the **Configuration** workspace.
    b. Expand the **Incidents** folder.
    c. Select **SNMP Trap Configurations** .
    d. Do one of the following:
        i. To create an incident configuration, click the ✱ New icon, and continue.
        ii. To edit an incident configuration, select a row, click the Open icon, and continue.
        iii. To delete an incident configuration, select a row and click the ✖ Delete icon.
2. Configure the required Basic settings (see table).
3. Click **Save and Close** to save your changes.

Basics Attributes for SNMP Trap Configuration

Task	How
"Specify the Incident Configuration	Use the **Basics** pane of the **SNMP Trap**

Basics Attributes for SNMP Trap Configuration, continued

Task	How
Name (SNMP Trap Incident)" on page 787	**Configuration** form. Specify a name that helps you to identify the configuration for subsequent use.
"Specify the SNMP Object ID" on page 788	Use the **Basics** pane of the **SNMP Trap Configuration** form. NNMi supports SNMPv3, SNMPv2c and SNMPv1 formats.
Specify whether you want to enable this configuration.	In the **Basics** group of the **SNMP Trap Configuration** form, verify that **Enable** ☑ is selected for each configuration you want to use.
"Display an SNMP Trap or an NNM 6.x/7.x Event as a Root Cause Incident" on page 789	Use the **Basics** pane of the **SNMP Trap Configuration** form.
"Specify Category and Family Attribute Values for Organizing Your Incidents (SNMP Trap Incident)" on page 790	Use the **Basics** pane of the **SNMP Trap Configuration** form. You can organize your incidents using Category and Family.
"Specify the Incident Severity (SNMP Trap Incident)" on page 794	Use the **Basics** pane of the **SNMP Trap Configuration** form. Possible Severity values include: **Normal, Warning, Minor, Major,** and **Critical**.
"Specify Your Incident Message Format (SNMP Trap Incident)" on page 795	Use the **Basics** pane of the **SNMP Trap Configuration** form. The message format determines the message to be displayed for the incident.
"Specify a Description for Your Incident Configuration (SNMP Trap Incident)" on page 802	Use the **Basics** pane of the **SNMP Trap Configuration** form. Provide a meaningful description.
Specify an Author for Your Incident Configuration (SNMP Trap Incident)	Use the **Basics** pane of the **SNMP Trap Configuration** form to indicate who created or last modified the trap. **Caution:** If the Author attribute value is **HP Network Node Manager**, any changes are at risk of being overwritten in the future. • Click ▾ **Lookup** and select **Show Analysis** to display details about the currently selected Author. • Click **Quick Find** to access the list of existing Author values. • Click ✱ **New** to create an Author value.

After you complete the Basic Configuration for the SNMP trap, you can also choose to configure the information described in the following table.

Additional Incident Configurations

Task	How
"Configure Interface Settings for an SNMP Trap Incident" on page 803	Select the **Interface Settings** tab to specify an Interface Group to which you want your incident configuration to apply.
"Configure Node Settings for an SNMP Trap Incident" on page 840	Select the **Node Settings** tab to specify a Node Group to which you want your incident configuration to apply.
"Configure Suppression Settings for an SNMP Trap Incident" on page 879	Select the **Suppression** tab to specify the criteria for discarding incidents that match the selected incident configuration.
"Configure Enrichment Settings for an SNMP Trap Incident" on page 887	Select the **Enrichment** tab to specify enhancements for the selected incident configuration.
"Configure Dampening Settings for an SNMP Trap Incident" on page 892	Select the **Dampen** tab to specify the time interval that must be met before the incident appears in an Incident view.
"Correlate Duplicate Incidents (Deduplication Configuration)" on page 659	Select the **Deduplication** tab to specify duplicate incidents that you want to be suppressed.
"Track Incident Frequency (Rate: Time Period and Count) " on page 659	Select the **Rate** tab to specify a rate for duplicate incidents. After the rate limit is reached, NNMi generates an Incident to notify you of the problem.
"Configure an Action for an Incident" on page 748	Select the **Actions** tab to specify actions that should occur automatically when an incident changes its Lifecycle State.

Specify the Incident Configuration Name (SNMP Trap Incident)

When providing the Name for an incident configuration, use the following guidelines:

Name
The name is used to identify the incident configuration and must be unique. Use a name that will help you to remember the purpose or kind of management event, remote NNM 6.x or 7.x event, or SNMP trap for which you are configuring this incident. Name is also used to identify your Pairwise configurations.

Specify the SNMP Object ID

When configuring incidents for an SNMP trap, you are asked to provide the SNMP Object ID values that you want to use to assist you in identifying the trap.

The SNMP Object IDs must be entered in a format that is recognized by NNMi. Select the type of SNMP trap you want to configure from the list below to learn about the required NNMi format.

Note: In all cases, the value you enter for an SNMP Object ID must be unique.

- "SNMP Object ID Format for SNMPv2c\SNMPv3 Traps" below
- "SNMP Object ID Format for SNMPv1 Generic Traps" below
- "SNMP Object ID Format for a Specific SNMPv1 Trap" on the next page

SNMPv1 generic traps are standard traps that are commonly used across vendors. The SNMPv1 enterprise specific traps are those traps that are generated by a particular vendor's device. The vendor is also known as the enterprise. Both include a vendor name as part of the set of information stored with each trap.

SNMP Object ID Format for SNMPv2c\SNMPv3 Traps

NNMi requires that all SNMP traps have an object identifier (SNMP Object ID).

To specify an SNMP trap object ID, open the MIB definition file for the device of interest to look up the correct ID. The MIB file includes object identifiers for all of the traps that the configured SNMP agent (SNMPv1, SNMPv2c, or SNMPv3) generates for a particular device.

In the **SNMP Object ID** attribute of the **SNMP Trap Configuration** form or **Remote NNM 6.x/7.x Event Configuration** form, enter the **SNMP Object ID** attribute value for the trap that you want to see in the console incident views.

SNMP Object ID Format for SNMPv1 Generic Traps

NNMi requires that SNMPv1 traps have object IDs. The object IDs are created according to the specifications in Request for Comments (RFC) document 2576: *Coexistence between Version 1, Version 2, and Version 3 of the Internet-standard Network Management Framework*.

When using SNMPv1 format, you can specify either generic or vendor specific traps. SNMPv1 generic traps are standard traps that are commonly used across vendors. The SNMPv1 enterprise specific traps are those traps that are generated by a particular vendor's device. The vendor is also known as the enterprise. Both include a vendor name as part of the set of information stored with each trap.

The six SNMPv1 generic traps have the following SNMP object identifiers that are recognized by SNMPv2c:

1.3.6.1.6.3.1.1.5.1 (coldStart)

1.3.6.1.6.3.1.1.5.2 (warmStart)

1.3.6.1.6.3.1.1.5.3 (linkDown)

1.3.6.1.6.3.1.1.5.4 (linkUp)

1.3.6.1.6.3.1.1.5.5 (authenticationFailure)

1.3.6.1.6.3.1.1.5.6 (egpNeighborLoss)

To configure an SNMP object identifier (SNMP OID) for a generic SNMPv1 trap, specify the SNMP object ID as described in RFC 2576. You also need to include the object identifier for the vendor name (<VendorEnterprise>) as shown below:

<*SNMPv2c generic trap OID*>.<*VendorEnterprise*>

The <*vendorEnterprise*> is the object identifier for the vendor that is included with the varbind trap information.

For example, the SNMP object identifier for Cisco warmStart trap would be:

```
.1.3.6.1.6.3.1.1.5.2.1.3.6.1.4.1.9
```

Note: Cisco's Vendor enterprise object identifier in this example is `.1.3.6.1.4.1.9`

SNMP Object ID Format for a Specific SNMPv1 Trap

NNMi requires that SNMPv1 traps have object identifiers. The object IDs are created according to the specifications in RFC 2576: *Coexistence between Version 1, Version 2, and Version 3 of the Internet-standard Network Management Framework*.

When using SNMPv1 format, you can specify either generic or vendor specific traps. SNMPv1 generic traps are standard traps that are commonly used across vendors. The SNMPv1 enterprise specific traps are those traps that are generated by a particular vendor's device. The vendor is also known as the entrprise. Both include a vendor name as part of the set of information stored with each trap.

When specifying the SNMP object ID for an SNMPv1 specific trap, include the SNMP object ID for the vendor name and for the trap that you want to see in the console incident views.

The value you enter must be in the format:

<*VendorEnterprise*>.0.<*SpecificTrapNumber*>

The <*VendorEnterprise*> is the object identifier for the vendor that is included in the SNMPv1 trap. The <*SpecificTrapNumber*> is the SNMPv1 specific trap identification number that is provided by the vendor.

For example, for an SNMPv1 vendor object id 1.3.6.1.3.1.12.9 and specific trap number 12234, the SNMP object ID would be:

```
1.3.6.1.3.1.12.9.0.12234
```

Display an SNMP Trap or an NNM 6.x/7.x Event as a Root Cause Incident

SNMP trap and NNM 6.x/7.x events normally appear as symptoms rather than as root cause incidents. However, there might be times when you want an SNMP or NNM 6.x/7.x event to appear as a root cause incident. For example, you might want an HSRP state change (cHsrpStateChange, 1.3.6.1.4.1.9.9.106.2.0.1) trap to be listed as a root cause. This trap might occur when the hot standby has gone down indicating the system is at risk if there is a failover.

Note: To reduce "noise" associated with duplicate incidents, NNMi changes the incident Correlation Nature to **Symptom** for any user-defined Root Cause incidents that exceed the rate or deduplication threshold.

To display an SNMP trap or NNM 6.x/7.x Event as a root cause incident:

Select **Root Cause** ☑ in the **SNMP Trap** or **Remote NNM 6.x/7.x Event Configuration** form.

The next time the incident occurs, the Correlation Nature attribute value changes to **Root Cause**.

To no longer display an SNMP trap or NNM 6.x/7.x Event s a root cause incident:

Clear **Root Cause** □ in the **SNMP Trap** or **Remote NNM 6.x/7.x Event Configuration** form.

The next time the incident occurs, the Correlation Nature attribute value changes to **Symptom**.

Specify Category and Family Attribute Values for Organizing Your Incidents (SNMP Trap Incident)

When configuring incidents, NNMi provides the Category and Family attributes to help you organize your incidents.

Preconfigured Categories
The Category attribute helps you organize your incidents. Select the category that you want to be associated with this type of incident when it appears in an incident view. Each of the possible Category values is described in the following table.

Incident Categories Provided by NNMi

Category	Description
Accounting	Used to indicate problems related to usage statistics and allocation of costs associated with the billing of time and services provided by devices. This category is not used by NNMi with default configurations, but it is available for incidents you define.
Application Status	Indicates there is a problem with the health of the NNMi software. Examples of these kinds of events include license expiration (see "Extend a Licensed Capacity" on page 1575) or that a certain NNMi process or service lost connection to the Process Status Manager (see "Stop or Start an NNMi Process" on page 82 and "Stop or Start NNMi Services" on page 86).
Configuration	Indicates there is a problem with the configuration of a managed device. For example, there is a physical address mismatch.
Fault	Indicates a problem with the network, for example Node Down.
Performance	Indicates a Monitored Attribute value *crossed* a configured threshold. For example, Disk Space Utilization exceeds the configured threshold criteria for High Value = 90 percent .
Security	Indicates there is a problem related to authentication. For example, an SNMP authentication failure.
Status	Indicates some kind of status message. Examples of these kinds of incidents include "SNMP Link Up" or an "HSRP Group status Normal" message.

Note: You can add your own Category entries to NNMi. See "Create an Incident Category (SNMP Trap Incident)" on the next page for more information.

You can use Family values to further categorize the types of incidents that might be generated. Each of the possible Family values are described in the following table.

Incident Family Attribute Values Provided by NNMi

Family	Description
Address	Indicates the incident is related to an address problem.
Aggregated Port	Indicates the incident is related to a **Link Aggregation**[1] problem.
BGP	Indicates the incident is related to a problem with BGP (Border Gateway Protocol). This family is not used by NNMi with default configurations, but it is available for incidents you define.
Board	Indicates the incident is related to a board problem. This family is not used by NNMi with default configurations, but it is available for incidents you define.
Chassis	Indicates the incident is related to an board problem. This family is not used by NNMi with default configurations, but it is available for incidents you define.
Component Health	Indicates the incident is related to Node Component metrics collected by NNMi. See "Node Form: Node Component Tab" for more information about the Node Component metrics collected.
Connection	Indicates the incident is related to a problem with one or more connections.
Correlation	Indicates the incident has additional incidents correlated beneath it. These incidents are associated with a duplicate count so that you can determine the number of correlated incidents associated with it.
Custom Poller	Indicates the incident is related to the NNMi Custom Poller feature. See About Custom Poller.
HSRP	*NNMi Advanced*. Indicates the incident is related to a problem with Hot Standby Router Protocol (**HSRP**[2]).
Interface	Indicates the incident is related to a problem with one or more interfaces.
License	Indicates the incident is related to a licensing problem.
NNMi	Indicates the incident is related to NNMi Health. See the Check NNMi Health for

[1] Protocols used on Switches to configure multiple Interfaces (Aggregation Member Interfaces) to function as if they were one (an Aggregator Interface). When two Aggregator Interfaces establish a connection, that connection is an Aggregator Layer 2 Connection. The Aggregator Layer 2 Connection appears on Layer 2 Neighbor View maps as a thick line with an Interface icon at each end (representing the Aggregator Interface). When you double-click the thick line, it converts into multiple thin lines representing the participating Aggregation Member Layer 2 Connections with their Aggregation Member Interfaces at each end of the lines.

[2] Hot Standby Router Protocol

Incident Family Attribute Values Provided by NNMi, continued

Family	Description
Health	more information.
Node	Indicates the incident is related to a node problem.
OSPF	Indicates the incident is related to an OSPF problem. This family is not used by NNMi with default configurations, but it is available for incidents you define.
RAMS	*NNMi Advanced*. Indicates the incident is related to a Router Analytics Management System problem.
RMON	Indicates the incident is related to a Remote Monitor (IETF standard, RFC 1757) problem. This family is not used by NNMi with default configurations, but it is available for incidents you define.
RRP	*NNMi Advanced*. Indicates the incident is related to a problem with a Router Redundancy Protocol configuration.
STP	Indicates the incident is related to Spanning-Tree Protocol problem. This family is not used by NNMi with default configurations, but it is available for incidents you define.
Syslog	NNMi does not use this Family with default configurations. It is available for incidents you define.
Trap Analysis	Indicates the incident is related to an SNMP trap storm.
VLAN	Indicates the incident is related to a problem with a virtual local area network.
VRRP	*NNMi Advanced*. Indicates the incident is related to a problem with Virtual Router Redundancy Protocol (**VRRP**[1]).

Note: You can add your own Family entries to NNMi. See "Create an Incident Family (SNMP Trap Incident)" on the next page for more information.

Create an Incident Category (SNMP Trap Incident)

The Category attribute helps you organize your incidents. Create any Category that makes sense to you and your team. For a list of the Category codes provided by NNMi, "Specify Category and Family Attribute Values for Organizing Your Incidents (SNMP Trap Incident)" on page 790.

To create a new incident Category:

1. Navigate to the **Incident Category** form.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Expand the **Incidents** folder.

[1] Virtual Router Redundancy Protocol

c. Select **SNMP Trap Configurations** .

d. Do one of the following:

   - To create an incident configuration, click the ✱ New icon.
   - To edit an incident configuration, double-click the row representing the configuration you want to edit.

e. In the configuration form, locate the **Category** attribute.

f. Click the Lookup icon, and select ✱ New.

2. Provide the required information (see table).

3. Click **Save and Close** to save your changes and return to the previous form.

Category Code Attributes

Name	Description
Label	Incident category name. For example, Hardware Faults, or Cisco Error. Maximum size is 255 characters. Alpha-numeric, spaces, and underline characters are permitted.

Unique Key	**Caution**: After you click **Save and Close**, this value cannot be changed.
	Used as a unique identifier when exporting and importing configuration definitions. To ensure that the value you enter is unique, it is recommended that you use the Java name space convention when providing this value. It is also useful to include the label value as part of the unique key as shown in the following examples:
	`com.<your_company_name>.nnm.trap_conf.category.<category_label>`
	`com.<your_company_name>.nnm.event_conf.category.<category_label>`
	`com.<your_company_name>.nnm.inci_conf.category.<category_label>`
	The maximum length is 80 characters. Alpha-numeric characters and periods are permitted. Spaces are not permitted.

Create an Incident Family (SNMP Trap Incident)

The Family attribute helps you organize your incidents. Create any Family that makes sense to you and your team. For a list of the Family codes provided by NNMi, "Specify Category and Family Attribute Values for Organizing Your Incidents (SNMP Trap Incident)" on page 790.

To create a new incident Family:

1. Navigate to the **Incident Family** form.

   a. From the workspace navigation panel, select the **Configuration** workspace.

   b. Expand the **Incidents** folder.

c. Select **SNMP Trap Configurations**.

d. Do one of the following:

- To create an incident configuration, click the ✱ New icon.
- To edit an incident configuration, double-click the row representing the configuration you want to edit.

e. In the configuration form, locate the **Family** attribute.

f. Click the Lookup icon, and select ✱ New.

2. Provide the required information (see table).

3. Click **Save and Close** to save your changes and return to the previous form.

Family Attributes

Name	Description
Label	Family name. For example, Hardware Faults, or Cisco Error. Maximum size is 255 characters. Any character type is valid.

Unique Key	**Caution**: After you click **Save and Close**, this value cannot be changed.
	Used as a unique identifier when exporting and importing configuration definitions. To ensure that the value you enter is unique, it is recommended that you use the Java name space convention when providing this value. It is also useful to include the label value as part of the unique key as shown in the following examples:
	`com.<your_company_name>.nnm.trapConf.family.<family_label>`
	`com.<your_company_name>.nnm.eventConf.family.<family_label>`
	`com.<your_company_name>.nnm.inciConf.family.<family_label>`
	The maximum length is 80 alpha-numeric characters, periods allowed, no spaces allowed.

Specify the Incident Severity (SNMP Trap Incident)

The incident severity represents the seriousness calculated for the incident. Use the severity attribute to specify the severity that should be assigned to the incident you are configuring. Possible values are described in the following table.

Incident Severity Values

Attribute	Description
Normal	Indicates there are no known problems related to the associated object. This severity is meant to be informational. Generally, no action is needed for these incidents.
Warning	Indicates there might be a problem related to the associated object.
Minor	Indicates NNMi has detected problems related to the associated object that require

Incident Severity Values, continued

Attribute	Description
	further investigation.
Major	Indicates NNMi has detected problems related to the associated object to be resolved before they become critical.
Critical	Indicates NNMi has detected problems related to the associated object that require immediate attention.

See "Monitor Incidents for Problems" for more information about these severity values.

Specify Your Incident Message Format (SNMP Trap Incident)

When configuring an incident, specify the information you want NNMi to include in the incident's Message attribute value. You can use any combination of valid parameter strings and Custom Incident attributes to configure the Message.

Note: The incident Message limit is 1024 characters. If the returned values exceed this limit, NNMi truncates the value starting from the end of the returned text string.

"Valid Parameters for Configuring Incident Messages (SNMP Trap Incident)" below

"Include Custom Incident Attributes in Your Message Format (SNMP Trap Incident)" on page 801

Valid Parameters for Configuring Incident Messages (SNMP Trap Incident)

When configuring incident messages, consider using incident information as part of the message. NNMi provides the following parameter values. Use these parameters as variables when formatting an incident message.

Tip: See the Using the Incident Form for more information about the parameter values.

Note: NNMi stores varbind values as custom incident attributes (CIAs).

Tip: If a value is not stored for a parameter, it is returned as “null”.

See "Specify Your Incident Message Format (SNMP Trap Incident)" above for more information about configuring messages.

Parameter strings are available for the following:

Note: See the following tables to view the valid parameters for incidents generated from Custom Polled Instances: Parameter Strings for all Incidents (Attributes from an Incident form), Parameter Strings for Node Source Objects (Attributes from a Node form), and the Parameter Strings for all Incidents (Attributes not Visible from any form).

- Parameter strings for all incidents (Incident form attributes) (Click here for a list of choices.)

Parameter Strings for all Incidents (Incident form attributes)

Parameter String	Description
$category, $cat	Value of the Category attribute in the Incident form.
$count, $cnt	Value representing the number of Custom Incident Attributes that appear in the Incident form.
$family, $fam	Value from the Family attribute in the Incident form.
$firstOccurrenceTime, $fot	Value from the First Occurrence Time attribute in the incident form.
$lastOccurrenceTime, $lot	Value from the Last Occurrence Time attribute in the incident form.
$lifecycleState, $lcs	Value from the Lifecycle State attribute in the Incident form.
$name	Value of the Name attribute from the incident configuration.
$nature, $nat	Value from the Nature attribute in the Incident form.
$origin, $ori	Value from the Origin attribute in the Incident form.
$originOccurrenceTime, $oot	Value from the Origin Occurrence Time attribute in the incident form.
$priority, $pri	Value from the Priority attribute in the Incident form.
$sev, $severity	Value of the Severity attribute of the Incident form.

- Parameter Strings for Node Source Objects (Node form attributes) (Click here for a list of choices.)

Parameter Strings for Node Source Objects (Node form attributes)

Parameter String	Description
$managementAddress, $mga	Value from the Management Address attribute of the incident's source Node's form or SNMP Agent form.
$otherSideOfConnectionManagementAddress, $oma	If the incident's Source Node is part of a Layer 2 Connection, this attribute is the value of the Management Address of a node on the other side of the Layer 2 Connection.
$sourceNodeLongName, $sln	The fully-qualified DNS name as displayed in the Hostname attribute of the incident's source Node's form.
$sourceNodeName, $snn	Value from the Name attribute of the incident's source Node's form.

Parameter Strings for Node Source Objects (Node form attributes) , continued

Parameter String	Description
$sysContact, $sct	Value from the System Contact attribute of the incident's source Node form: General tab.
$sysLocation, $slc	Value from the System Location attribute of the incident's source Node form: General tab.

- Parameter Strings for Interface Source Objects (Interface form attributes) (Click here for a list of choices.)

Parameter Strings for Interface Source Objects (Interface form attributes)

Parameter String	Description
$ifAlias, $ifa	Value from the IfAlias attribute for the interface that is the incident's source object.
$ifConfigDupSetting, $icd	Configured Duplex Setting on the port associated with the interface that is the incident's source object.
$ifDesc, $idc	Value from the ifDesc attribute for the interface that is the incident's source object.
$ifIndex, $idx	Value from the ifIndex attribute for the interface that is the incident's source object.
$ifIpAddr, $iia	IP Address values associated with the interface that is the incident's source object. If multiple IPaddresses are associated with the interface, this parameter returns a comma-separated list.
$ifName, $ifn	Value from the ifName attribute for the interface that is the incident's source object.
$ifPhysAddr, $ipa	Value from the Physical Address attribute for the interface that is the incident's source object.
$ifSpeed, $isp	Value from the ifSpeed attribute for the interface that is the incident's souce object.
$ifType, $itp	Value from the ifType attribute for the interface that is the incident's souce object.

- Parameter Strings for Layer 2 Connection Source Objects (Layer 2 Connection form attributes) (Click here for a list of choices.)

Parameter Strings for Layer 2 Connection Source Objects (Layer 2 Connection form attributes)

Parameter String	Description
$otherSideOfConnectionConfigDupSetting, $ocd	If the incident's source Node is part of a Layer 2 Connection, this parameter contains the Configured Duplex Setting on the port associated with the interface on the other side of the connection.
$otherSideOfConnectionIfAlias, $oia	If the incident's Source Node is part of a Layer 2 Connection, this parameter is the value of the ifAlias of one of the interfaces on the other side of the Layer 2 Connection.
$otherSideOfConnectionIfDesc, $odc	If the incident's Source Node is part of a Layer 2 Connection, this parameter contains the ifDescr attribute value for the interface on the other side of the Layer 2 Connection.
$otherSideOfConnectionIfIndex, $odx	If the incident's Source Node is part of a Layer 2 Connection, this parameter contains the ifIndex attribute value for the interface on the other side of the connection.
$otherSideOfConnectionIfName, $ofn	If the incident's Source Node is part of a Layer 2 Connection, this parameter contains the ifName attribute value for the interface on the other side of the connection.

- Parameter strings for VLAN Source Objects (VLAN form attributes) (Click here for a list of choices.)

Parameter Strings for VLAN Source Objects (VLAN form attributes)

Parameter String	Description
$impVlanIds, $ivi	Value from the VLAN Id attribute associated with the interface that is the incident's source object. To access this information from an interface form, navigate to the VLAN Port tab and open the form for the VLAN of interest. If the interface is part of more than one VLAN, this parameter returns a comma-separated list.
$impVlanNames, $ivn	Value from the VLAN Name attribute associated with the interface that is the incident's source object. To access this information from an interface form, navigate to the VLAN Ports tab of the Interface form. If the interface is part of more than one VLAN, this parameter returns a comma-separated list.

- Parameter Strings for all incidents (Additional information that is not visible in any form) (Click here for a list of choices.)

Parameter Strings for all Incidents (Attributes not visible in any form)

Parameter String	Description
$firstOccurrenceTimeMs, $fms	Value from the First Occurrence Time attribute in the incident form, converted to millseconds (measured since January 1, 1970, 00:00:00 GMT - Greenwich Mean Time).
$lastOccurrenceTimeMs, $lms	Value from the Last Occurrence Time attribute in the incident form, converted to millseconds (measured since January 1, 1970, 00:00:00 GMT - Greenwich Mean Time).
$oid	Value of the unique object identifier (oid) for the incident configuration that originated from either an SNMP trap, Management Event, or Remote NNM 6.x or 7.x event.
$otherSideOfConnection, $osc	If the incident's Source Node is part of a Layer 2 Connection, this attribute is the following combination of values for the node and one of its interfaces on the other side of the Layer 2 Connection: The fully-qualified DNS name of the node appended with the interface Name in the following format: <*fully-qualified DNS name*>[*interface_name*]
$originOccurrenceTimeMs $oms	Value from the Origin Occurrence Time attribute in the incident form, converted to millseconds (measured since January 1, 1970, 00:00:00 GMT - Greenwich Mean Time).
$sourceNodeUuid, $snu	Universally Unique Object Identifier attribute value of the source node object for the incident (unique across all databases). This identifier distinguishes the source node object instance from all other node objects.
$sourceObjectClass, $soc	Value of the object class for the object you want to include. Use this parameter to request more details of a class of objects through a web service. Examples of object classes include: `com.hp.ov.nms.model.core.Interface` and `com.hp.ov.nms.model.snmp.SnmpAgent`.
$sourceObjectName, $son	Value from the Name attribute of the source object. For example, an interface object is named according to the MIB ifName. Each ifName varies according to the vendor's conventions. Using the name `4/1` as an example, `4` represents `the board number and 1` represents the port number.
$sourceObjectUuid, $sou	Universally Unique Object Identifier attribute value of the source object for the incident (unique across all databases). This identifier distinguishes the source object instance from all other similar object instances.
$uuid	Universally Unique Object Identifier attribute value of the

Parameter Strings for all Incidents (Attributes not visible in any form), continued

Parameter String	Description
	incident (unique across all databases). This identifier distinguishes the incident object instance from all other incident objects.

- Information established in Custom Incident Attributes (Click here for a list of choices.)

Parameter Strings for Attributes Established in Custom Incident Attributes

Parameter String	Description
$<position_number>	Value of the custom incident attribute (CIA) position number for any CIA that originated from a varbind or was added by NNMi. For example, to indicate you want to use the varbind in position 1, enter: `$1` NNMi stores varbind values as Custom Incident Attributes. If you know the varbind position number, use this parameter.
$<CIA_ name>	Value of the name that is used for the custom incident attribute. For example, `$mycompany.mycia`. NNMi provides CIA values for configuring Management Events. See Custom Incident Attributes Provided by NNMi for more information about custom incident attributes.
$<CIA_ oid>	Value of the object identifier for any custom incident attribute that originated as a varbind. For example, `$.1.3.6.1.6.3.1.1.5.1`. Use this parameter when you are not certain of a custom incident attribute (varbind) position number.
$*	Used to indicate you want all of the custom incident attribute values originating as varbinds, to be passed to the action configuration. Each varbind is returned in the following format: `$<CIA_name>:<CIA_value>` in which the custom incident attribute name appears followed by the custom incident attribute value.

- Functions to generate values (Click here for a list of choices.)

  The function described in the following table replaces the specified numeric value with the associated text value stored in the CIA.

  Note: The associated MIB must have been loaded using the nnmloadmib.ovpl command.

Functions to Generate Values Within the Incident Message

Function	Description
$oidtext ($<position_ number>)	A *<position_number>* argument specifies the numeric value of the custom incident attribute (CIA) position number for any CIA that originated from a varbind or was added by NNMi. For example, `$oidtext($2)`. **Note**: The position number you enter must represent a CIA that contains an Object Identifier (OID) value. NNMi returns the textual value of the OID for the CIA specified.

Functions to Generate Values Within the Incident Message, continued

Function	Description
	Note the following: ▪ If the MIB is not loaded, NNMi returns the numeric OID value. ▪ If the OID has a MIB instance, the number representing the MIB instance is appended to the textual OID value.
$oidtext ($<CIA_oid>)	The <*CIA_oid*> argument specifies the Object Identifier (OID) for any custom incident attribute that originated as a varbind. For example, `$oidtext ($.1.3.6.1.6.3.1.1.5.1.)` Use this argument to the $oidtext() function when you are not certain of a custom incident attribute (varbind) position number. NNMi replaces the numeric value with the textual value of the OID you specify. Note the following: ▪ If the MIB is not loaded, NNMi returns the numeric OID value. ▪ If the OID has a MIB instance, the number representing the MIB instance is appended to the textual OID value.
$text ($<position_number>)	The <*position_number*> argument specifies the numeric value of the custom incident attribute (CIA) position number for any CIA that originated from a varbind or was added by NNMi. For example, to indicate you want to use the varbind in position 1, enter: `$1`. NNMi replaces the numeric value with the text value stored in the CIA. **Note**: If a text value is not available, NNMi returns the numeric value.
$text ($<CIA_oid>)	The <CIA_oid> argument specifies the object identifier for any custom incident attribute that originated as a varbind. For example, `$.1.3.6.1.6.3.1.1.5.1.` Use this argument to the $text function when you are not certain of a custom incident attribute (varbind) position number. NNMi replaces the numeric value with the text value stored in the CIA. **Note**: If a text value is not available, NNMi returns the numeric value.

Include Custom Incident Attributes in Your Message Format (SNMP Trap Incident)

NNMi includes two categories of CIAs:

- SNMP trap varbinds identified by the Abstract Syntax Notation value (ASN.1). NNMi turns varbinds into CIAs and maintains each varbind's position number. See "Load SNMP Trap Incident Configurations" on page 771.
- Custom incident attributes provided by NNMi. See "Custom Incident Attributes Provided by NNMi (Information for Administrators)" on page 647.

  You cannot create Custom Incident Attributes.

You can use CIAs in your message format to extend the amount of information presented. To determine which CIAs are available for any particular incident type, open an Incident view, locate the incident and open the Incident form. Navigate to the **Custom Attributes** tab. A complete list of available CIAs (for that incident type) appears in the table.

To include a CIA in your message format, type the dollar-sign character (`$`) plus any of the following:

- Varbind position number or asterisk (*) to include all varbind values
- Name of the CIA
- Object identifier (oid) of the CIA (useful when the varbind position number is not consistent among vendors)

Note: A single incident cannot include two CIAs with the same name. However, two incidents can contain CIAs having the same names and values.

The following table presents some example formats with the subsequent output.

Example Incident Message Formats

Example Message Format	Output in Incident View
Possible trouble with $3	`Possible trouble with` <varbind 3>
Possible trouble with $11	`Possible trouble with` <varbind 11>
Possible trouble with $77 (where the varbind position 77 does not exist)	`Possible trouble with <Invalid or unknown cia> 77`
Possible trouble with $*	`Possible trouble with` <cia1_name: cia_value>, <cia2_name; cia_value>,< cia*n*_name: cia_value>
Possible trouble with $3x	`Possible trouble with` <varbind 3>`x`
Possible trouble with $1.2.3.4.5	`Possible trouble with` <value of the CIA with oid of 1.2.3.4.5>
Possible trouble with $mycia.mycompany	`Possible trouble with` <value of the CIA with name of mycia.mycompany>

Tip: NNMi provides an error message when a CIA cannot be found. For example, if you enter an unavailable varbind position, name, or object identifier (oid), NNMi returns an "Invalid or unknown cia" error message.

Specify a Description for Your Incident Configuration (SNMP Trap Incident)

NNMi provides the Description attribute to help you further identify the current incident configuration.

Description

Use the description field to provide additional information that you would like to store about the current incident configuration. This description applies only to the configuration entry.

Type a maximum of 1024 characters. Alpha-numeric, spaces, and special characters (~ ! @ # $ % ^ & * ( ) _+ -) are permitted.

Configure Interface Settings for an SNMP Trap Incident

NNMi enables you to apply a Suppression, Enrichment, Dampen, or Actions incident configuration to a Source Object based on the Source Object's participation in an Interface Group.

Note: Interface Settings override any other Suppression, Enrichment, Dampen, or Actions configuration settings for this incident, including those configured on the Node Settings tab.

Tip: See "Create Interface Groups" on page 321 for more information about Interface Groups.

For information about each Interface Settings tab:

For information about each SNMP Traps tab:

To apply an incident configuration to a Source Object based on the Source Object's Interface Group:

1. Navigate to the **SNMP Trap Configuration** form:
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Expand the **Incidents** folder.
   c. Select **SNMP Trap Configurations** .
   d. Do one of the following:
      i. To create an incident configuration, click the New icon, and continue.
      ii. To edit an incident configuration, select a row, click the Open icon, and continue.
      iii. To delete an incident configuration, select a row, and click the Delete icon.
2. Select the **Interface Settings** tab.
3. Do one of the following:
   a. To create a new configuration, click the New icon.
   b. To edit an existing configuration, click the Open icon in the row representing the configuration you want to edit.
4. Configure the desired Interface Settings (see table).
5. Configure any Suppression, Dampen, or Enrichment settings for this Interface Group.
6. Click **Save and Close** to save your changes and return to the previous form.

Interface Group Attributes

Name	Description
Interface Group	Click the Lookup icon and select Quick Find to select the Interface Group you want to use. See "Use the Quick Find Window" on page 41 for more information about using Quick Find.
Ordering	Determines the priority order for those interfaces that appear in multiple Interface Groups. The lower the number, the higher the priority. For example, **1** is the highest

Interface Group Attributes , continued

Name	Description
	priority. If an interface is in multiple Interface Groups and more than one of those Interface Groups have been specified in an incident configuration, only the incident configuration with the highest priority will be applied to the interface.
Enable	Use this attribute to temporarily disable an incident's configuration settings: **Enable** □ = Temporarily disable the selected configuration. **Enable** ☑ = Enable the selected configuration.

Related Topics

"Configure Node Settings for an SNMP Trap Incident" on page 840

Configure Incident Suppression Settings for an Interface Group (SNMP Trap Incident)

Note: Interface Settings override any other Suppression settings for this incident, including those from the Node Settings tab.

NNMi enables you to suppress a specified incident configuration based on the Source Object's participation in an Interface Group. When an incident is suppressed:

- It is not stored in the NNMi database
- It does not appear in an incident view in the NNMi console

You can also suppress the incident configuration based on either of the following:

- Source Node's participation in a Node Group. See "Configure Incident Suppression Settings for a Node Group (SNMP Trap Incident)" on page 841 for more information.
- Incident configuration default settings without specifying a Node or Interface Group. See "Configure Suppression Settings for an SNMP Trap Incident" on page 879 for more information.

Tip: See "Create Interface Groups" on page 321 for more information about Interface Groups.

For information about each Interface Settings tab:

To suppress an incident configuration based on an Interface Group:

1. Navigate to the **SNMP Trap Configuration** form:
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Expand the **Incidents** folder.
   c. Select **SNMP Trap Configurations** .
   d. Do one of the following:
      i. To create an incident configuration, click the ✱ New icon, and continue.

   ii. To edit an incident configuration, select a row, click the Open icon, and continue.

   iii. To delete an incident configuration, select a row, and click the Delete icon.

2. Navigate to the **Interface Settings** tab.

3. Do one of the following:
   a. To create a new configuration, click the * New icon.

   b. To edit an existing configuration, select a row, and click the Open icon.

4. Make sure you configure the basic Interface Setting behavior. See "Configure Interface Settings for an SNMP Trap Incident" on page 803 for more information.

5. Select the **Suppression** tab.

6. Configure the desired Suppression behavior (see table).

7. Click **Save and Close** to save your changes and return to the previous form.

Interface Settings Suppression Attributes

Name	Description
Enable	Use this attribute to temporarily disable an incident's suppression settings for the specified Interface Group: **Enable** □ = Temporarily disable the selected configuration. **Enable** ☑ = Enable the selected configuration.
Payload Filter	The Payload Filter Editor enables you to create expressions that further refine the filters used to select the incidents to be suppressed, enriched, or dampened. Make sure to design any complex Payload Filters offline as a Boolean expression first. This method can help to minimize errors when entering your expressions using the Payload Filter editor. When creating a Payload Filter, note the following: • Payload Filter expressions for the `like` and `not like` operators use the syntax defined for java regular expressions (java.util.regex Pattern class). • You must use a `ciaName` that already exists in the trap or event you are configuring. • Each set of expressions associated with a Boolean Operator is treated as if it were enclosed in parentheses and evaluated together. • View the expression displayed under **Filter String** to see the logic of the expression as it is created. • The `AND` and `OR` Boolean Operators must contain at least two expressions as shown in the example below. The following example filters incidents on voltage state: `AND`

Interface Settings Suppression Attributes , continued

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td></td><td>

```
ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7
ciaValue = 5
```

NNMi evaluates the expression above as follows:

`(ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7 AND ciaValue = 5)`

NNMi finds all incidents with a varbind `.1.3.6.1.4.1.9.9.13.1.2.1.7` value of **5**.

- The placement of your cursor and the subsequent text that is selected is important when performing operations using the Payload Filter Editor. For example, you append to, replace, or change the indentation of the expression that is selected.
- The placement of your cursor and the subsequent text that is selected is especially important when adding your Boolean operators.
- You can include more than one varbind in the same Payload Filter expression as shown in the following example:

```
((ciaName like \Q.1.3.6.1.4.1.9.9\E.* AND ciaValue = 25) AND
(ciaName like \Q.1.3.6.1.2.1.2.2.1.1.3\E.* AND ciaValue = 3))
```

In this example, a given trap must meet each of the following criteria:

- Contain a varbind whose Object Identifier (OID) matches the regular expression `\Q.1.3.6.1.4.1.9.9\E.*` and has a value of `25`.
- Contain a varbind whose OID matches the regular expression `\Q.1.3.6.1.2.1.2.2.1.1.3\E.*` and has a value of `3`.

Payload Filter Editor Components

<table>
<tr><th>Attribute</th><th>Description</th></tr>
<tr><td>Attribute</td><td>The attribute name on which NNMi searches. Filterable attributes include the following:

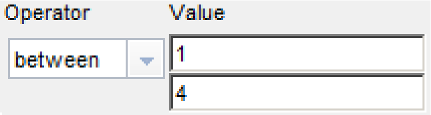
- ciaName
- ciaValue
</td></tr>
<tr><td>Operator</td><td>Valid operators are described below.

- **=** Finds all values equal to the value specified. Click here for an example.

  Example: `ciaName=.1.3.6.1.4.1.9.9.13.1.2.1.7` matches any incident that contains a varbind with the name value of **.1.3.6.1.4.1.9.9.13.1.2.1.7**.
- **!=** Finds all values not equal to the value specified. Click here for an example.
</td></tr>
</table>
</td></tr>
</table>

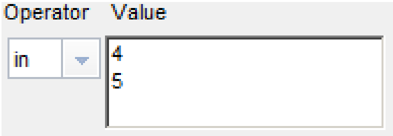
Interface Settings Suppression Attributes , continued

Name	Description
	Payload Filter Editor Components, continued

Attribute	Description
	Example: `ciaName! = .1.3.6.1.4.1.9.9.13.1.2.1.7` matches any incident that contains a varbind with a name value other than **1.3.6.1.4.1.9.9.13.1.2.1.7**. • **<** Finds all values less than the value specified. Click here for an example. Example: `ciaValue < 6` matches any incident that contains a varbind with a value less than **6**. • **<=** Finds all values less than or equal to the value specified. Click here for an example. Example: `ciaValue <= 6` matches any incident that contains a varbind with a value less than or equal to **6**. • **>** Finds all values greater than the value specified. Click here for an example. Example: `ciaValue > 4` matches any incident that contains a varbind with a value greater than **4**. • **>=** Finds all values greater than or equal to the value specified. Click here for an example. Example: `ciaValue >= 4` matches any incident that contains a varbind with values greater than or equal to **4**. • **between** Finds all values equal to and between the two values specified. Click here for an example. Example: `ciaValue between`  Operator: between Value: 1 4 matches any incident that contains a varbind value equal to or greater than **1** and equal to or less than **4**. **Note**: As shown in the example, each value must be entered on a separate line. • **in** Finds any match to at least one value in a list of values. Click here for an example. Example: `ciaValue in`

Interface Settings Suppression Attributes , continued

Name	Description
	Payload Filter Editor Components, continued

Attribute	Description
	 Operator: in Value: 4 5 matches any incident with a varbind value of either **4** or **5**. **Note**: As shown in the example, each value must be entered on a separate line. NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example (**4, 5**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line. - **is not null** Finds all non-blank values. Click here for an example. Example: `ciaValue is not null` matches any incident with a varbind that contains a value. - **is null** Finds all blank values. Click here for an example. Example: `ciaValue is null` matches any incident with a varbind that does not contain a value. - **like** Finds matches using the syntax defined for java regular expressions. See the Pattern (Java Platform SE6) API documentation at : `http://download.oracle.com/javase/6/docs/api/java/util/regex/Pattern.html` for more information. Click here for more information. The period asterisk (.*) characters mean *any number of characters of any type at this location*. The period (.) character means *any single character of any type at this location*. **Note**: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below. Example: `ciaName like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that begins with 1.3.6.1.4.1.9.9 and (optionally) ends with any number of characters. `ciaValue like .*Chicago.*` finds all traps or events that contain a

Interface Settings Suppression Attributes , continued

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td></td><td>

Payload Filter Editor Components, continued

<table>
<tr><th>Attribute</th><th>Description</th></tr>
<tr><td></td><td>

varbind value that includes the string **Chicago**.

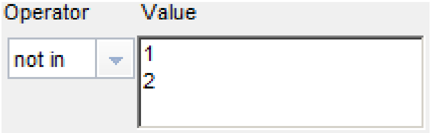
- **not between** Finds all values except those between the two values specified. Click here for an example.

  Example: `ciaValue not between 5 8` matches an incident that contains a varbind with the values less than **5** or greater than **8** .

- **not in** Finds all values except those included in the list of values. Click here for an example.

  Example:

  `ciaValue not in`

  Operator: not in
  Value:
  1
  2

  matches any incident that contains a varbind with values other than **1** and **2**.

  Note: As shown in the example, each value must be entered on a separate line.

  NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example, (**1, 2**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line.

- **not like** Finds all that do not have the values specified using the syntax defined for Java regular expressions. See the Pattern (Java Platform SE6) API documentation at: `http://download.oracle.com/javase/6/docs/api/java/util/regex/Pattern.html` for more information. Click here for an example.

  The period asterisk (.*) characters mean *any number of characters of any type at this location*.

  The period (.) character means *any single character of any type at this location*.

  Note: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below.

</td></tr>
</table>

</td></tr>
</table>

Interface Settings Suppression Attributes , continued

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td></td><td>
<b>Payload Filter Editor Components, continued</b>
<table>
<tr><th>Attribute</th><th>Description</th></tr>
<tr><td></td><td>Example:
<code>ciaName not like \Q.1.3.6.1.4.1.9.9\E.*</code> matches any incident that contains a varbind name value that does not begin with <b>.1.3.6.1.4.1.9.9</b> and (optionally) ends with any number of characters.
<code>ciaValue not like .*Chicago.*</code> finds all traps or events that do not contain a varbind value that includes the string <b>Chicago</b>.</td></tr>
<tr><td>Value</td><td>The value for which you want NNMi to search.
Note the following:
• The values you enter are case sensitive.
• NNMi displays a variable number of value fields depending on the Operator selected. For example, the <code>between</code> Operator causes two value fields to be displayed.
• The <code>between, in</code> and <code>not in</code> operators require that each value be entered on a separate line.</td></tr>
</table>
<b>Payload Filter Editor Buttons</b>
<table>
<tr><th>Button</th><th>Description</th></tr>
<tr><td>Append</td><td>Appends the current expression (Attribute, Operator,and Value) to the selected expression already included in the filter string.</td></tr>
<tr><td>Insert</td><td>Inserts the current expression (Attribute, Operator,and Value) in front of the cursor location within the Filter String.</td></tr>
<tr><td>Replace</td><td>Replaces the selected expression with the expression displayed in the Attribute, Operator, and Value fields.</td></tr>
<tr><td>AND</td><td>Inserts the AND Boolean Operator in the selected cursor location.
<b>Note</b>: View the expression displayed under <b>Filter String</b> to see the logic of the expression as it is created.</td></tr>
<tr><td>OR</td><td>Inserts the OR Boolean Operator in the current cursor location.
<b>Note</b>: View the expression displayed under <b>Filter String</b> to see the logic of the expression as it is created.</td></tr>
<tr><td>NOT</td><td>Can be used in any part of the Filter String to specify that NNMi should</td></tr>
</table>
</td></tr>
</table>

Interface Settings Suppression Attributes , continued

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td></td><td>

Payload Filter Editor Buttons, continued

Button	Description
	exclude interfaces with values that pass the expression that immediately follows the `NOT`. For example, when evaluating the following expression, NNMi includes interfaces with (interface description) `ifDesc` containing **VLAN**, and excludes any Interfaces that have **VLAN10** for the (interface name) `ifName` value: `(ifDesc like VLAN AND NOT (ifName=VLAN10))` **Note**: View the expression displayed under Filter String to see the logic of the expression as it is created .
EXISTS	Used for filters that include Capabilities or Custom Attribute names and values in the Filer String. Indicates that you want NNMi to consider interfaces that do not have any Capabilities or Custom Attributes when evaluating the Filter String. **Note**: If you include Capabilities or Custom Attribute names and values in the Filer String, but do not use **EXISTS** or **NOT EXISTS**, NNMi excludes from its search interfaces that do not include Capabilities or Custom Attributes. For example, when evaluating the following Filter String, NNMi includes interfaces with (interface description) `ifDesc` containing **VLAN**, as well as any Interfaces Custom Attribute Role value is **LAN Connection to Oracle Server**: `(ifDesc like VLAN OR EXISTS((customAttrName=Role AND customAttrValue=LAN Connection to Oracle Server)))` **Note**: View the expression displayed under Filter String to see the logic of the expression as it is created.
NOT EXISTS	Used for filters that include Capabilities or Custom Attribute names and values in the Filer String. Indicates that you want NNMi to consider interfaces that do not have any Capabilities or Custom Attributes when evaluating the Filter String, but exclude the interfaces that match the expression that follows the **NOT EXISTS**. **Note**: If you include Capabilities or Custom Attribute names and values in the Filer String, but do not use **EXISTS** or **NOT EXISTS**, NNMi excludes from its search interfaces that do not

</td></tr>
</table>

Interface Settings Suppression Attributes , continued

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td></td><td>
<b>Payload Filter Editor Buttons, continued</b>
<table>
<tr><th>Button</th><th>Description</th></tr>
<tr><td></td><td>include Capabilities or Custom Attributes.
For example, when evaluating the following expression, NNMi includes interfaces with (interface description) <code>ifDesc</code> containing <b>VLAN</b>, and excludes any Interfaces that have the Custom Attribute Role and that Role value is <b>LAN Connection to Oracle Server</b>:
<code>(ifDesc like VLAN OR NOT EXISTS ((customAttrName=Role AND customAttrValue=LAN Connection to Oracle Server)))</code>
<b>Note</b>: View the expression displayed under Filter String to see the logic of the expression as it is created.</td></tr>
<tr><td>Delete</td><td>Deletes the selected expression.
<b>Note</b>: If the Boolean Operator is selected, the Payload Filter Editor deletes all expressions associated with the Boolean Operator.</td></tr>
</table>
</td></tr>
</table>

Configure Incident Enrichment Settings for an Interface Group (SNMP Trap Incident)

Note: Interface Settings override any other Enrichment settings for this incident, including those from the Node Settings tab.

NNMi enables you to fine tune and enhance a specified incident configuration based on the Source Object's participation in an Interface Group. The types of items you can fine tune and enhance for a selected incident configuration, include:

- Severity
- Priority
- Category
- Family
- Correlation Nature
- Message
- Assigned To

You can also enrich the incident configuration based on either of the following:

- The incident configuration based on the Source Node's participation in a Node Group. See "Configure Incident Enrichment Settings for a Node Group (SNMP Trap Incident)" on page

849 for more information.

- Incident configuration default settings without specifying a Node or Interface Group. See "Configure Enrichment Settings for an SNMP Trap Incident" on page 887 for more information.

Tip: See Create Interface Groups for more information about Interface Groups.

For information about each Interface Settings tab:

For information about each Enrichment tab:

To enrich an incident configuration based on an Interface Group:

1. Navigate to the **SNMP Trap Configuration** form:
   1. From the workspace navigation panel, select the **Configuration** workspace.
   2. Expand the **Incidents** folder.
   3. Select **SNMP Trap Configurations** .
   4. Do one of the following:
      1. To create an incident configuration, click the New icon, and continue.
      2. To edit an incident configuration, select a row, click the Open icon, and continue.
      3. To delete an incident configuration, select a row, and click the Delete icon.
2. Select the **Interface Settings** tab.
3. Do one of the following:
   1. To create a new configuration, click the New icon.
   2. To edit an existing configuration, select a row, click the Open icon, and continue.
4. Make sure you configure the basic Interface Setting behavior. See "Configure Interface Settings for an SNMP Trap Incident" on page 803 for more information.
5. Select the **Enrichment** tab.
6. Do one of the following:
   1. To create an Enrichment configuration, click the New icon and continue.
   2. To edit an Enrichment configuration, select a row, click the Open icon, and continue.
   3. To delete an Enrichment configuration, select a row and click the Delete icon.
7. Configure the desired Enrichment behavior (see the "Enrich Configuration Attributes" table)
8. Click **Save and Close** to save your changes and return to the previous form.

Interface Settings Enrich Configuration Attributes

Name	Description
Category	Use the Category attribute to customize the category for this incident configuration. Possible values include:

Interface Settings Enrich Configuration Attributes , continued

Name	Description
	• Accounting • Application Status • Configuration • Fault • Performance • Security • Status See "Specify Category and Family Attribute Values for Organizing Your Incidents (SNMP Trap Incident)" on page 790 for more information.
Family	Use the Family attribute to customize the Family for this incident configuration. Select from the drop-down list or create a new value. For example, some of the values provided by NNMi include: • Address • Aggregated Port (Interfaces using **Link Aggregation**[1] protocol. See Interface Form: Link Aggregation tab.) • Card • Connection • Correlation • Interface • Node
Severity	The incident Severity represents the seriousness calculated for the incident. Use the Severity attribute to specify the severity that should be assigned to the incident you are configuring. Possible values are described below: **Normal** - Indicates there are no known problems related to the associated object. This Severity is meant to be informational. Generally, no action is needed for these incidents. **Warning** - Indicates there might be a problem related to the associated object. **Minor** - Indicates NNMi has detected problems related to the associated object

[1] Protocols used on Switches to configure multiple Interfaces (Aggregation Member Interfaces) to function as if they were one (an Aggregator Interface). When two Aggregator Interfaces establish a connection, that connection is an Aggregator Layer 2 Connection. The Aggregator Layer 2 Connection appears on Layer 2 Neighbor View maps as a thick line with an Interface icon at each end (representing the Aggregator Interface). When you double-click the thick line, it converts into multiple thin lines representing the participating Aggregation Member Layer 2 Connections with their Aggregation Member Interfaces at each end of the lines.

Interface Settings Enrich Configuration Attributes , continued

Name	Description
	that require further investigation. **Major** - Indicates NNMi has detected problems related to the associated object to be resolved before they become critical. **Critical** - Indicates NNMi has detected problems related to the associated object that require immediate attention.
Priority	Used to communicate the urgency of resolving the selected incident. You control this value. NNMi sets this value to null by default. The lower the number the higher the priority. Possible values are: 5 **None** 4 **Low** 3 **Medium** 2 **High** 1 **Top** **Note**: The icons are displayed only in table views.
Correlation Nature	Use the Correlation Nature to customize the Correlation Nature for this incident configuration. Possible values include: • Info • None • Root Cause • Secondary Root Cause • Symptom • Stream Correlation • Service Impact • Dedup Stream Correlation • Rate Stream Correlation See Incident Form: General Tab for more information.
Message Format	When configuring an incident, specify how the incident message appears in the incident view. The string you specify in the Message Format attribute is visible in an incident view. **Note**: The incident message limit is 1024 characters. If you exceed this limit, NNMi truncates the value starting from the right.

Interface Settings Enrich Configuration Attributes , continued

Name	Description
	You can use any combination of default and custom attributes: "Valid Parameters for Configuring Incident Messages (SNMP Trap Incident)" on page 795 "Include Custom Incident Attributes in Your Message Format (SNMP Trap Incident)" on page 801
Assigned To	Use to specify the owner of any incident generated for this incident configuration. Click the Lookup icon and select Quick Find to select a valid user name. **Note**: You can also begin to type a valid user name and use the auto-complete feature to select the user name of interest.
Description	Use the Description attribute to provide additional information that you want to note about the current enhancement configuration. This description applies only to the enhancement configuration and does not appear when NNMi displays any associated incident. Type a maximum of 1024 characters. Alpha-numeric and special characters (~ ! @ # $ % ^ & * ( ) _+ -) are permitted.

Configure Custom Incident Attributes to Enrich an Incident Configuration (Interface Settings) (SNMP Trap Incidents)

The Custom Incident Attributes (CIAs) tab enables you to specify additional CIAs to be saved with an incoming incident. The persisted data might then be used as an argument to an action for the incident.

When creating a CIA for an incident configuration, you can specify any of the following values:

- Custom Attribute on the source node
- Custom Attribute on the interface (source object)

You also specify the Custom Incident Attribute name that will store this information.

For information about each Enrichment tab:

To create a Custom Incident Attribute to enrich an incident configuration:

1. Navigate to the **SNMP Trap Configuration** form:
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Expand the **Incidents** folder.
   c. Select **SNMP Trap Configurations** .
   d. Do one of the following:

i. To create an incident configuration, click the New icon, and continue.

ii. To edit an incident configuration, select a row, click the Open icon, and continue.

iii. To delete an incident configuration, select a row, and click the Delete icon.

2. Select the **Interface Settings** tab.

3. Do one of the following:

   a. To create a new configuration, click the New icon.

   b. To edit an existing configuration, double-click the row representing the configuration you want to edit.

4. Make sure you configure the basic Interface Setting behavior. See "Configure Interface Settings for an SNMP Trap Incident" on page 803 for more information.

5. Select the **Enrichment** tab.

6. Do one of the following:

   a. To create an Enrichment configuration, click the New icon, and continue.

   b. To edit an Enrichment configuration,select a row, click the Open icon, and continue.

   c. To delete an Enrichment configuration, select a row and click the Delete icon.

5. Make sure the Enrichment settings are configure. See "Configure Incident Enrichment Settings for an Interface Group (SNMP Trap Incident)" on page 812 for more information.

8. Navigate to the **Custom Incident Attributes** tab.

9. Do one of the following:

   a. To create a Custom Incident Attribute, click the New icon, and continue.

   b. To edit a Custom Incident Attribute, select a row, click the Open icon, and continue.

   c. To delete a Custom Incident Attribute, select a row and click the Delete icon.

10. Configure the Custom Incident Attribute (see the "Custom Incident Attribute" table).

11. Click **Save and Close** to save your changes and return to the previous form.

Custom Incident Attribute

Name	Description
Type	Specifies whether you are using a Custom Attribute on a node or a Custom Attribute on an interface. Possible values are: • Node Custom Attribute • Interface Custom Attribute
Custom Attribute Name	Used to determine the value to be assigned to the Custom Incident Attribute you are configuring. Enter either of the following: • Name of the Custom Attribute on the source node • Name of the Custom Attribute on the interface (source object)

Custom Incident Attribute , continued

Name	Description
Custom Incident Attribute Name	Name used to identify the Custom Incident Attribute you are configuring. The name limit is 255 characters. Alpha-numeric characters are permitted. No spaces or special characters (~ ! @ # $ % ^ & * ( ) _+ -) are permitted.

Configure a Payload Filter to Enrich an Incident Configuration (Interface Settings) (SNMP Trap Incidents)

The Payload Filter Editor enables you to create expressions that further refine the filters used to select the incidents to be enriched. Make sure to design any complex Payload Filters offline as a Boolean expression first. This method can help to minimize errors when entering your expressions using the Payload Filter editor.

To create a Payload Filter expression:

1. Navigate to the **SNMP Trap Configuration** form:
    a. From the workspace navigation panel, select the **Configuration** workspace.
    b. Expand the **Incidents**folder.
    c. Select **SNMP Trap Configurations**.
    d. Do one of the following:
        i. To create an incident configuration, click the ✳ New icon, and continue.
        ii. To edit an incident configuration, select a row, click the 📂 Open icon, and continue.
        iii. To delete an incident configuration, select a row and click the ✖ Delete icon.
2. Select the **Interface Settings** tab.
3. Do one of the following:
    a. To create a new configuration, click the ✳ New icon.
    b. To edit an existing configuration, select a row, click the 📂 Open icon, and continue..
4. Make sure you configure the basic Interface Setting behavior. See "Configure Interface Settings for an SNMP Trap Incident" on page 803 for more information.
5. Select the **Enrichment** tab.
6. Do one of the following:
    a. To create an Enrichment configuration, click the ✳ New icon, and continue.
    b. To edit an Enrichment configuration, select a row, click the 📂 Open icon, and continue.
    c. To delete an Enrichment configuration, select a row and click the ✖ Delete icon.

7. Make sure the Enrichment settings are configured. See "Configure Incident Enrichment Settings for an Interface Group (SNMP Trap Incident)" on page 812 for more information.

8. Select the **Payload Filter** tab.

9. Define your Payload Filter (see table). Also see "Guidelines for Creating a Payload Filter".

   a. Plan out the logic needed for your Filter String.

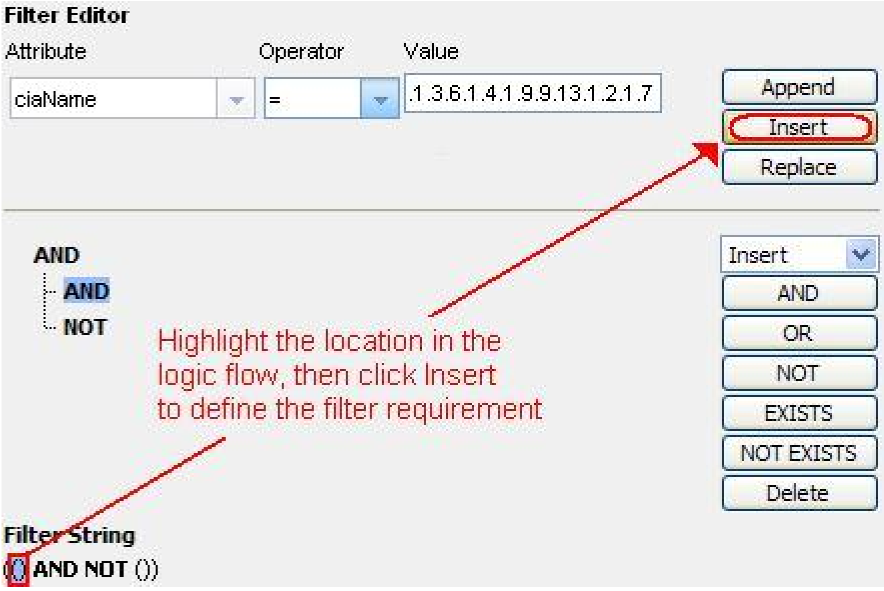
   b. Use the buttons on the bottom half of the Additional Filters Editor to establish the logic structure.

      For example, to establish the following structure, select **Insert**, then click **AND**, then **NOT**, and then **AND** a second time:

      ```
      (( ) AND NOT ( ))
      ```

   c. Now place your cursor in a location within the displayed Filter String, and use the top half of the filter editor to define the parameters of the highlighted filter requirement.

      For example, select a set of parentheses and use the Insert button to specify the filter requirement within those parentheses:



Filter Editor

Attribute: ciaName

Operator: =

Value: .1.3.6.1.4.1.9.9.13.1.2.1.7

Append
Insert
Replace
AND
AND
NOT
Insert
AND
OR
NOT
EXISTS
NOT EXISTS
Delete
Highlight the location in the logic flow, then click Insert to define the filter requirement

Filter String

(() AND NOT ())

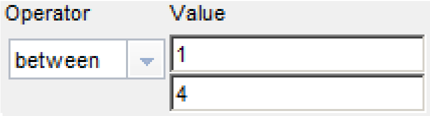
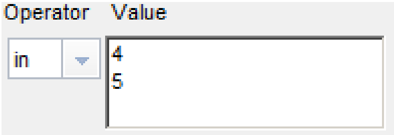
10. Click **Save and Close**.

11. Click **Save and Close** to save your changes and return to the previous form.

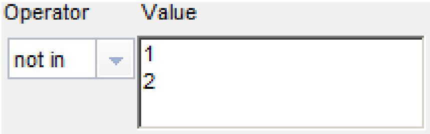
Payload Filter Editor Components

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: • ciaName • ciaValue
Operator	Valid operators are described below.

Payload Filter Editor Components, continued

Attribute	Description
	• **=** Finds all values equal to the value specified. Click here for an example. Example: `ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7` matches any incident that contains a varbind with the name value **.1.3.6.1.4.1.9.9.13.1.2.1.7**. • **!=** Finds all values not equal to the value specified. Click here for an example. Example: `ciaName != .1.3.6.1.4.1.9.9.13.1.2.1.7` matches any incident that contains a varbind with the name value other than **1.3.6.1.4.1.9.9.13.1.2.1.7**. • **<** Finds all values less than the value specified. Click here for an example. Example: `ciaValue < 6` matches any incident with a varbind value less than **6**. • **<=** Finds all values less than or equal to the value specified. Click here for an example. Example: `ciaValue <= 6` matches any incident with a varbind value less than or equal to **6**. • **>** Finds all values greater than the value specified. Click here for an example. Example: `ciaValue > 4` matches any incident with a varbind value greater than **4**. • **>=** Finds all values greater than or equal to the value specified. Click here for an example. Example: `ciaValue >= 4` matches any incident that contains a varbind with values greater than or equal to **4**. • **between** Finds all traps or events that include a varbind with a value equal to and between the two values specified. Click here for an example. Example: `ciaValue between`  Operator: between Value: 1 4 matches any incident that contains a varbind value equal to or greater than **1** and equal to or less than **4**. **Note**: As shown in the example, each value must be entered on a separate line. • **in** Finds any match to at least one value in a list of values. Click here for an example. Example: `ciaValue in`  Operator: in Value: 4 5

Payload Filter Editor Components, continued

Attribute	Description
	matches any incident with a varbind value of either **4** or **5**. **Note**: As shown in the example, each value must be entered on a separate line. NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example (**4, 5**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line. • **is not null** Finds all non-blank values. Click here for an example. Example: `ciaValue is not null` matches any incident with a varbind that contains a value. • **is null** Finds all blank values. Click here for an example. Example: `ciaValue is null` matches any incident with a varbind that does not have a value. • **like** Finds matches using wildcard characters. Click here for more information about using wildcard characters. The period asterisk (.*) characters mean *any number of characters of any type at this location*. The period (.) character means *any single character of any type at this location*. **Note**: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below. Examples: `ciaName like  \Q.1.3.6.1.4.1.9.9\E.*` finds all traps or events that contain varbind names that begin with **.1.3.6.1.4.1.9.9** and (optionally) end with any number of characters. `ciaValue like .*Chicago.*` finds all traps or events that contain a varbind value that includes the string **Chicago**. • **not between** Finds all values except those between the two values specified. Click here for an example. Example: `ciaValue not between 5 8` matches an incident that contains a varbind with the values less than **5** or greater than **8**. • **not in** Finds all values except those included in the list of values. Click here for an example. Example: `ciaValue not in`  Operator: not in Value: 1 2 matches any incident that contains a varbind with values other than **1** and **2**.

Payload Filter Editor Components, continued

Attribute	Description
	Note: As shown in the example, each value must be entered on a separate line. NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example, (**1, 2**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line. • **not like** Finds all that do not have the values specified (using wildcard strings). Click here for an example. The period asterisk (.*) characters mean *any number of characters of any type at this location*. The period (.) character means *any single character of any type at this location*. **Note**: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below. Example: `ciaName not like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that does not begin with **.1.3.6.1.4.1.9.9**. `ciaValue not like .*Chicago.*` finds all traps or events that do not contain a varbind value that includes the string **Chicago**.
Value	The value for which you want NNMi to search. Note the following: • The values you enter are case sensitive. • NNMi displays a variable number of value fields depending on the Operator selected. For example, the `between` Operator causes two value fields to be displayed. • The `between, in` and `not in` operators require that each value be entered on a separate line.

Additional Filters Editor Buttons

Button	Description
Append	Appends the current expression (Attribute, Operator,and Value) to the selected expression already included in the Filter String.
Insert	Inserts the current expression (Attribute, Operator,and Value) in front of the cursor location within the Filter String.
Replace	Replaces the selected expression with the expression displayed in the Attribute, Operator, and Value fields.
AND	Appends, inserts, or replaces the AND Boolean Operator in the selected cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the

Additional Filters Editor Buttons, continued

Button	Description
	expression as it is created.
OR	Appends, Inserts, or replaces the OR Boolean Operator in the current cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
NOT	Can be used in any part of the Filter String to specify that NNMi should exclude interfaces with values that pass the expression that immediately follows the `NOT`. For example, when evaluating the following expression, NNMi includes interfaces with (interface description) `ifDesc` containing **VLAN**, and excludes any Interfaces that have **VLAN10** for the (interface name) `ifName` value: `(ifDesc like VLAN AND NOT (ifName=VLAN10))` **Note**: View the expression displayed under Filter String to see the logic of the expression as it is created .
EXISTS	Used for filters that include Capabilities or Custom Attribute names and values in the Filer String. Indicates that you want NNMi to consider interfaces that do not have any Capabilities or Custom Attributes when evaluating the Filter String. **Note**: If you include Capabilities or Custom Attribute names and values in the Filer String, but do not use **EXISTS** or **NOT EXISTS**, NNMi excludes from its search interfaces that do not include Capabilities or Custom Attributes. For example, when evaluating the following Filter String, NNMi includes interfaces with (interface description) `ifDesc` containing **VLAN**, as well as any Interfaces Custom Attribute Role value is **LAN Connection to Oracle Server**: `(ifDesc like VLAN OR EXISTS((customAttrName=Role AND customAttrValue=LAN Connection to Oracle Server)))` **Note**: View the expression displayed under Filter String to see the logic of the expression as it is created.
NOT EXISTS	Used for filters that include Capabilities or Custom Attribute names and values in the Filer String. Indicates that you want NNMi to consider interfaces that do not have any Capabilities or Custom Attributes when evaluating the Filter String, but exclude the interfaces that match the expression that follows the **NOT EXISTS**. Note: If you include Capabilities or Custom Attribute names and values in the Filer String, but do not use **EXISTS** or **NOT EXISTS**, NNMi excludes from its search interfaces that do not include Capabilities or Custom Attributes. For example, when evaluating the following expression, NNMi includes interfaces with (interface description) `ifDesc` containing **VLAN**, and excludes any Interfaces that have the Custom Attribute Role and that Role value is **LAN**

Additional Filters Editor Buttons, continued

Button	Description
	Connection to Oracle Server: `(ifDesc like VLAN OR NOT EXISTS((customAttrName=Role AND customAttrValue=LAN Connection to Oracle Server)))` **Note**: View the expression displayed under Filter String to see the logic of the expression as it is created.
Delete	Deletes the selected expression. **Note**: If the Boolean Operator is selected, the Additional Filters Editor deletes all expressions associated with the Boolean Operator.

Configure Incident Dampening Settings for an Interface Group (SNMP Trap Incident)

Note: Interface Settings override any other Dampening settings for this incident, including those from the Node Settings tab.

NNMi enables you to delay the following for an incident configuration based on the Source Object's participation in an Interface Group:

- Execution of Incident Actions
- Execution of Diagnostics (*HP Network Node Manager iSPI Network Engineering Toolset Software \ NNM iSPI NET*)
- Appearance within Incident views in the NNMi Console

You can configure the Dampening settings based on either of the following:

- The Source Node's participation in a Node Group. See "Configure Incident Dampening Settings for a Node Group (SNMP Trap Incident)" on page 861 for more information.
- Incident configuration default settings without specifying a Node or Interface Group. See "Configure Dampening Settings for an SNMP Trap Incident" on page 892 for more information.

Tip: See "Create Interface Groups" on page 321 for more information about Interface Groups.

For information about each Interface Settings tab:

After the Dampen Interval has passed, NNMi changes the Lifecycle State to REGISTERED.

See About the Incident Lifecycle for more information about Lifecycle State.

To configure Dampening settings based on an Interface Group:

1. Navigate to the **SNMP Trap Configuration** form:
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Expand the **Incidents** folder.
   c. Select **SNMP Trap Configurations** .

   d. Do one of the following:

      i. To create an incident configuration, click the New icon, and continue.

      ii. To edit an incident configuration, select a row, click the Open icon, and continue.

      iii. To delete an incident configuration, select a row and click the Delete icon.

2. Select the **Interface Settings** tab.

3. Do one of the following:
   a. To create a new configuration, click the New icon.

   b. To edit an existing configuration, select a row, click the Open icon, and continue.

4. Make sure you configure the basic Interface Setting behavior. See "Configure Interface Settings for an SNMP Trap Incident" on page 803 for more information.

5. Select the **Dampening** tab.

6. Configure the desired Dampening behavior (see table).

7. Click **Save and Close** to save your changes and return to the previous form.

Interface Settings Dampening Configuration Attributes

Name	Description
Enable	Use this attribute to temporarily disable an incident's dampening settings: **Enable** □ = Temporarily disable the selected configuration. **Enable** ☑ = Enable the selected configuration.
Hour	Specifies the number of hours to be used for the dampen interval.
Minutes	Specifies the number of minutes to be used for the dampen interval. **Note:** The maximum dampen interval is 60 minutes.
Seconds	Specifies the number of seconds to be used for the dampen interval.
Payload Filter	The Payload Filter Editor enables you to create expressions that further refine the filters used to select the incidents to be suppressed, enriched, or dampened. Make sure to design any complex Payload Filters offline as a Boolean expression first. This method can help to minimize errors when entering your expressions using the Payload Filter editor. When creating a Payload Filter, note the following: • Payload Filter expressions for the `like` and `not like` operators use the syntax defined for java regular expressions (java.util.regex Pattern class). • You must use a `ciaName` that already exists in the trap or event you are configuring. • Each set of expressions associated with a Boolean Operator is treated as if it were enclosed in parentheses and evaluated together.

Interface Settings Dampening Configuration Attributes , continued

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td></td><td>

- View the expression displayed under **Filter String** to see the logic of the expression as it is created.
- The `AND` and `OR` Boolean Operators must contain at least two expressions as shown in the example below.

  The following example filters incidents on voltage state:

  ```
  AND
      ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7
      ciaValue = 5
  ```

  NNMi evaluates the expression above as follows:

  ```
  (ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7 AND ciaValue = 5)
  ```

  NNMi finds all incidents with a varbind `.1.3.6.1.4.1.9.9.13.1.2.1.7` value of **5.**
- The placement of your cursor and the subsequent text that is selected is important when performing operations using the Payload Filter Editor. For example, you append to, replace, or change the indentation of the expression that is selected.
- The placement of your cursor and the subsequent text that is selected is especially important when adding your Boolean operators.
- You can include more than one varbind in the same Payload Filter expression as shown in the following example:

  ```
  ((ciaName like \Q.1.3.6.1.4.1.9.9\E.* AND ciaValue = 25) AND
  (ciaName like \Q.1.3.6.1.2.1.2.2.1.1.3\E.* AND ciaValue =
  3))
  ```

  In this example, a given trap must meet each of the following criteria:

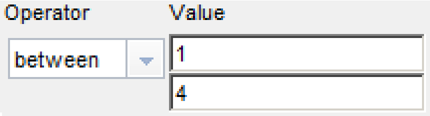
  - Contain a varbind whose Object Identifier (OID) matches the regular expression `\Q.1.3.6.1.4.1.9.9\E.*` and has a value of `25`.
  - Contain a varbind whose OID matches the regular expression `\Q.1.3.6.1.2.1.2.2.1.1.3\E.*` and has a value of `3`.

Payload Filter Editor Components

<table>
<tr><th>Attribute</th><th>Description</th></tr>
<tr><td>Attribute</td><td>The attribute name on which NNMi searches. Filterable attributes include the following:
• ciaName
• ciaValue</td></tr>
<tr><td>Oper</td><td>Valid operators are described below.</td></tr>
</table>

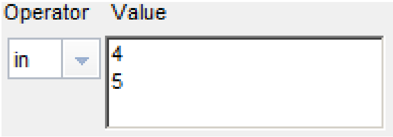
</td></tr>
</table>

Interface Settings Dampening Configuration Attributes , continued

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td></td><td>
<b>Payload Filter Editor Components, continued</b>
<table>
<tr><th>Attribute</th><th>Description</th></tr>
<tr><td>ator</td><td>
• <b>=</b> Finds all values equal to the value specified. Click here for an example.
Example: <code>ciaName=.1.3.6.1.4.1.9.9.13.1.2.1.7</code> matches any incident that contains a varbind with the name value of <b>.1.3.6.1.4.1.9.9.13.1.2.1.7</b>.
• <b>!=</b> Finds all values not equal to the value specified. Click here for an example.
Example: <code>ciaName! = .1.3.6.1.4.1.9.9.13.1.2.1.7</code> matches any incident that contains a varbind with a name value other than <b>1.3.6.1.4.1.9.9.13.1.2.1.7</b>.
• <b>&lt;</b> Finds all values less than the value specified. Click here for an example.
Example: <code>ciaValue &lt; 6</code> matches any incident that contains a varbind with a value less than <b>6</b>.
• <b>&lt;=</b> Finds all values less than or equal to the value specified. Click here for an example.
Example: <code>ciaValue &lt;= 6</code> matches any incident that contains a varbind with a value less than or equal to <b>6</b>.
• <b>&gt;</b> Finds all values greater than the value specified. Click here for an example.
Example: <code>ciaValue &gt; 4</code> matches any incident that contains a varbind with a value greater than <b>4</b>.
• <b>&gt;=</b> Finds all values greater than or equal to the value specified. Click here for an example.
Example: <code>ciaValue &gt;= 4</code> matches any incident that contains a varbind with values greater than or equal to <b>4</b>.
• <b>between</b> Finds all values equal to and between the two values specified. Click here for an example.
Example: <code>ciaValue between</code>

Operator: between
Value: 1
4
matches any incident that contains a varbind value equal to or greater than <b>1</b> and equal to or less than <b>4</b>.
<b>Note</b>: As shown in the example, each value must be entered on a separate line.
</td></tr>
</table>
</td></tr>
</table>

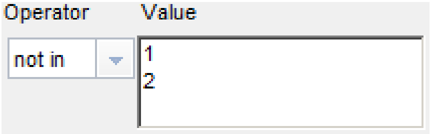
Interface Settings Dampening Configuration Attributes , continued

Name	Description
	Payload Filter Editor Components, continued

Attribute	Description
	• **in** Finds any match to at least one value in a list of values. Click here for an example. Example: `ciaValue in` Operator Value in 4 5 matches any incident with a varbind value of either **4** or **5**. **Note**: As shown in the example, each value must be entered on a separate line. NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example (**4, 5**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line. • **is not null** Finds all non-blank values. Click here for an example. Example: `ciaValue is not null` matches any incident with a varbind that contains a value. • **is null** Finds all blank values. Click here for an example. Example: `ciaValue is null` matches any incident with a varbind that does not contain a value. • **like** Finds matches using the syntax defined for java regular expressions. See the Pattern (Java Platform SE6) API documentation at : `http://download.oracle.com/javase/6/docs/api/java/util/regex/Pattern.html` for more information. Click here for more information. The period asterisk (.*) characters mean *any number of characters of any type at this location*. The period (.) character means *any single character of any type at this location*. **Note**: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below. Example:

Interface Settings Dampening Configuration Attributes , continued

Name	Description
	Payload Filter Editor Components, continued

Attribute	Description
	`ciaName like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that begins with 1.3.6.1.4.1.9.9 and (optionally) ends with any number of characters. `ciaValue like .*Chicago.*` finds all traps or events that contain a varbind value that includes the string **Chicago**. • **not between** Finds all values except those between the two values specified. Click here for an example. Example: `ciaValue not between 5 8` matches an incident that contains a varbind with the values less than **5** or greater than **8** . • **not in** Finds all values except those included in the list of values. Click here for an example. Example: `ciaValue not in`  Operator not in Value 1 2 matches any incident that contains a varbind with values other than **1** and **2**. **Note**: As shown in the example, each value must be entered on a separate line. NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example, (**1, 2**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line. • **not like** Finds all that do not have the values specified using the syntax defined for Java regular expressions. See the Pattern (Java Platform SE6) API documentation at: `http://download.oracle.com/javase/6/docs/api/java/util/regex/Pattern.html` for more information. Click here for an example. The period asterisk (.*) characters mean *any number of characters of any type at this location*. The period (.) character means *any single character of any type at this location*.

Interface Settings Dampening Configuration Attributes , continued

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td></td><td>

Payload Filter Editor Components, continued

<table>
<tr><th>Attribute</th><th>Description</th></tr>
<tr><td></td><td>

Note: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below.

Example:

`ciaName not like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that does not begin with **.1.3.6.1.4.1.9.9** and (optionally) ends with any number of characters.

`ciaValue not like .*Chicago.*` finds all traps or events that do not contain a varbind value that includes the string **Chicago**.

</td></tr>
<tr><td>Value</td><td>

The value for which you want NNMi to search.

Note the following:

- The values you enter are case sensitive.
- NNMi displays a variable number of value fields depending on the Operator selected. For example, the `between` Operator causes two value fields to be displayed.
- The `between, in` and `not in` operators require that each value be entered on a separate line.

</td></tr>
</table>

Payload Filter Editor Buttons

<table>
<tr><th>Button</th><th>Description</th></tr>
<tr><td>Append</td><td>Appends the current expression (Attribute, Operator,and Value) to the selected expression already included in the filter string.</td></tr>
<tr><td>Insert</td><td>Inserts the current expression (Attribute, Operator,and Value) in front of the cursor location within the Filter String.</td></tr>
<tr><td>Replace</td><td>Replaces the selected expression with the expression displayed in the Attribute, Operator, and Value fields.</td></tr>
<tr><td>AND</td><td>Inserts the AND Boolean Operator in the selected cursor location.
Note: View the expression displayed under **Filter String** to see the logic of the expression as it is created.</td></tr>
<tr><td>OR</td><td>Inserts the OR Boolean Operator in the current cursor location.
Note: View the expression displayed under **Filter String** to see the logic of the expression as it is created.</td></tr>
</table>

</td></tr>
</table>

Interface Settings Dampening Configuration Attributes , continued

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td></td><td>

Payload Filter Editor Buttons, continued

<table>
<tr><th>Button</th><th>Description</th></tr>
<tr><td>NOT</td><td>Can be used in any part of the Filter String to specify that NNMi should exclude interfaces with values that pass the expression that immediately follows the `NOT`.
For example, when evaluating the following expression, NNMi includes interfaces with (interface description) `ifDesc` containing **VLAN**, and excludes any Interfaces that have **VLAN10** for the (interface name) `ifName` value:
`(ifDesc like VLAN AND NOT (ifName=VLAN10))`
Note: View the expression displayed under Filter String to see the logic of the expression as it is created .</td></tr>
<tr><td>EXISTS</td><td>Used for filters that include Capabilities or Custom Attribute names and values in the Filer String.
Indicates that you want NNMi to consider interfaces that do not have any Capabilities or Custom Attributes when evaluating the Filter String.
Note: If you include Capabilities or Custom Attribute names and values in the Filer String, but do not use **EXISTS** or **NOT EXISTS**, NNMi excludes from its search interfaces that do not include Capabilities or Custom Attributes.
For example, when evaluating the following Filter String, NNMi includes interfaces with (interface description) `ifDesc` containing **VLAN**, as well as any Interfaces Custom Attribute Role value is **LAN Connection to Oracle Server**:
`(ifDesc like VLAN OR EXISTS((customAttrName=Role AND customAttrValue=LAN Connection to Oracle Server)))`
Note: View the expression displayed under Filter String to see the logic of the expression as it is created.</td></tr>
<tr><td>NOT EXISTS</td><td>Used for filters that include Capabilities or Custom Attribute names and values in the Filer String. Indicates that you want NNMi to consider interfaces that do not have any Capabilities or Custom Attributes when evaluating the Filter String, but exclude the interfaces that match the expression that follows the **NOT EXISTS**.
Note: If you include Capabilities or Custom Attribute names and values in the Filer String, but do not use **EXISTS** or **NOT EXISTS**, NNMi excludes from its search interfaces that do not</td></tr>
</table>

</td></tr>
</table>

Interface Settings Dampening Configuration Attributes , continued

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td></td><td>

Payload Filter Editor Buttons, continued

<table>
<tr><th>Button</th><th>Description</th></tr>
<tr><td></td><td>include Capabilities or Custom Attributes.
For example, when evaluating the following expression, NNMi includes interfaces with (interface description) <code>ifDesc</code> containing <b>VLAN</b>, and excludes any Interfaces that have the Custom Attribute Role and that Role value is <b>LAN Connection to Oracle Server</b>:
<code>(ifDesc like VLAN OR NOT EXISTS ((customAttrName=Role AND customAttrValue=LAN Connection to Oracle Server)))</code>
<b>Note</b>: View the expression displayed under Filter String to see the logic of the expression as it is created.</td></tr>
<tr><td>Delete</td><td>Deletes the selected expression.
<b>Note</b>: If the Boolean Operator is selected, the Payload Filter Editor deletes all expressions associated with the Boolean Operator.</td></tr>
</table>

</td></tr>
</table>

Configure Incident Actions for an Interface Group (SNMP Trap Incident)

Note: Interface Settings override any other Actions settings for this incident, including those from the Node Settings tab.

For information about each Interface Settings tab:

NNMi enables you to configure incident actions based on a Source Object's participation in an Interface Group.

You can also configure incident actions based on either of the following:

- The Source Node's participation in a Node Group. See "Configure Incident Actions for a Node Group (SNMP Trap Incident)" on page 869 for more information.
- Incident configuration default settings without specifying a Node or Interface Group. See "Configure Actions for an SNMP Trap Incident" on page 913 for more information.

You can configure actions to automatically run at any point in the incident lifecycle. For example, you might want to configure an action to occur when an incident of the type you are configuring is generated (**Registered**). When an incident is generated, you might want to automatically open a trouble ticket or send email or page your network operator. After the incident is **Closed**, you might want to automatically close the trouble ticket.

> **Note:** Your actions will not be executed until you enable the Actions configuration by either clicking Enable ☑ on the Actions tab or using the **Actions** → **Enable Configuration** option.

You can configure actions for incidents generated from SNMP Traps and the NNMi management events. Any time an incident configuration changes, the action directory is rescanned and any executable or script files (for example, Jython) are reloaded to the NNMi database. See "Lifecycle Transition Action Form (SNMP Trap Incidents)" on page 914 for more information about the actions directory.

Tip: Copy any required executable or script files to the NNMi actions directory before you configure an incident action. New or updated actions are loaded into NNMi only when an incident configuration is updated or created. See "Lifecycle Transition Action Form (SNMP Trap Incidents)" on page 914 for the location of the NNMi action directory.

When the action runs, output is logged to the `incidentActions.*.*.log` file. See "Verify that NNMi Services are Running" on page 85 for more information about log files and where they are located.

To configure an automatic action for an incident:

1. Navigate to the **SNMP Trap Configuration** tab.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Expand the **Incidents** folder.
   c. Select **SNMP Trap Configurations** .
   d. Do one of the following:
      i. To create a new incident configuration, click the New icon.
      ii. To edit an existing incident configuration, select a row, click the Open icon, and continue.
2. Select the **Interface Settings** tab.
3. Do one of the following:
   a. To create a new configuration, click the New icon.
   b. To edit an existing configuration, select a row, click the Open icon, and continue.
4. Make sure the basic Interface Setting behavior is configured. See "Configure Interface Settings for an SNMP Trap Incident" on page 803 for more information.
5. Select the **Actions** tab.
6. From the **Lifecycle Actions** table toolbar, do one of the following:
   - To create an Action configuration, click the New icon, and continue.
   - To edit an Action configuration, select a row, click the Open icon, and continue.
   - To delete an Action configuration, select a row and click the Delete icon.
7. In the "Lifecycle Transition Action Form (SNMP Trap Incidents)" on page 914, provide the required information.
8. Click **Save and Close** to save your changes.

   The next time the lifecycle changes, NNMi launches the action associated with the lifecycle for the incident of that type .

Configure a Payload Filter for an Incident Action (Interface Settings) (SNMP Trap Incidents)

The Payload Filter Editor enables you to create expressions that further refine the filters used to select the incidents that cause the configured action to run. Make sure to design any complex Payload Filters offline as a Boolean expression first. This method can help to minimize errors when entering your expressions using the Payload Filter editor.

To create a Payload Filter expression:

1. Navigate to the **SNMP Trap Configuration** form:
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Expand the **Incidents** folder.
   c. Select the **SNMP Traps** tab.
   d. Do one of the following:
      i. To create an incident configuration, click the New icon, and continue.
      ii. To edit an incident configuration, select a row, click the Open icon, and continue..
      iii. To delete an incident configuration, select the row and click the Delete icon.
2. Select the **Interface Settings** tab.
3. Do one of the following:
   a. To create a new configuration, click the New icon.
   b. To edit an existing configuration,select a row, click the Open icon, and continue.
4. Make sure you configure the basic Interface Setting behavior. See "Configure Interface Settings for an SNMP Trap Incident" on page 803 for more information.
5. Select the **Actions** tab.
6. Do one of the following:
   a. To create an Action configuration, click the New icon, and continue.
   b. To edit an Action configuration, select a row, click the Open icon, and continue.
   c. To delete an Action configuration, select a row and click the Delete icon.
7. Make sure you configure the Action Configuration settings. See "Configure Incident Actions for an Interface Group (SNMP Trap Incident)" on page 832 for more information.

   h. Select the **Payload Filter** tab.

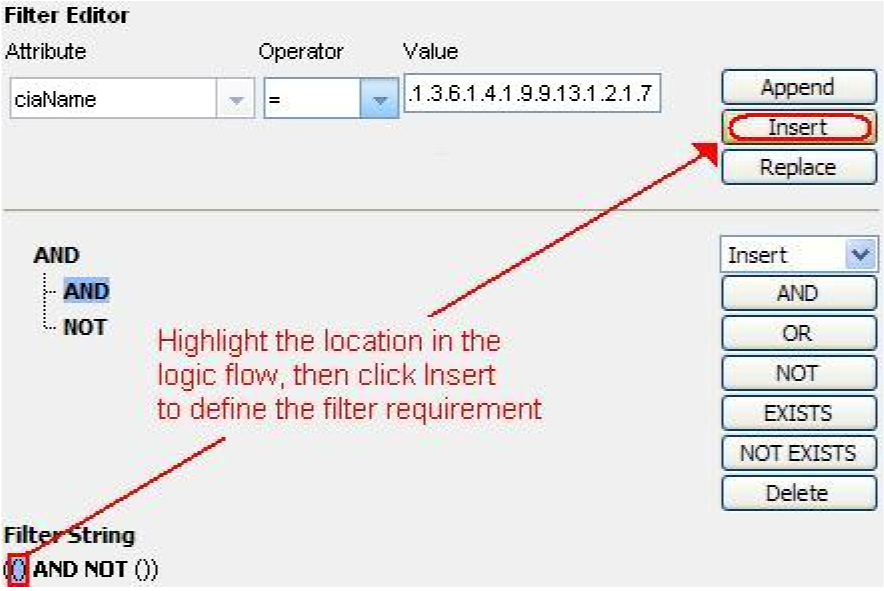
   i. Define your Payload Filter (see table). Also see "Guidelines for Creating a Payload Filter".
      i. Plan out the logic needed for your Filter String.
      ii. Use the buttons on the bottom half of the Additional Filters Editor to establish the logic structure.

For example, to establish the following structure, select **Insert**, then click **AND**, then **NOT**, and then **AND** a second time:

```
(( ) AND NOT ( ))
```

iii. Now place your cursor in a location within the displayed Filter String, and use the top half of the filter editor to define the parameters of the highlighted filter requirement.

For example, select a set of parentheses and use the Insert button to specify the filter requirement within those parentheses:



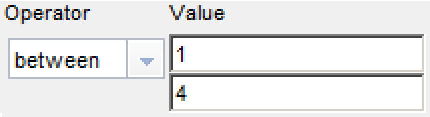
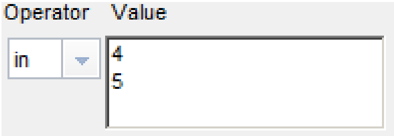
j. Click Save and Close.

k. Click Save and Close to save your changes and return to the previous form.

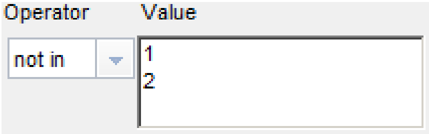
Payload Filter Editor Components

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: ▪ ciaName ▪ ciaValue
Operator	Valid operators are described below. ▪ **=** Finds all values equal to the value specified. Click here for an example. Example: `ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7` matches any incident that contains a varbind with the name value **.1.3.6.1.4.1.9.9.13.1.2.1.7**. ▪ **!=** Finds all values not equal to the value specified. Click here for an example. Example: `ciaName != .1.3.6.1.4.1.9.9.13.1.2.1.7` matches any incident that contains a varbind with the name value other than

Payload Filter Editor Components, continued

Attribute	Description
	1.3.6.1.4.1.9.9.13.1.2.1.7. - **<** Finds all values less than the value specified. Click here for an example. Example: `ciaValue < 6` matches any incident with a varbind value less than **6**. - **<=** Finds all values less than or equal to the value specified. Click here for an example. Example: `ciaValue <= 6` matches any incident with a varbind value less than or equal to **6**. - **>** Finds all values greater than the value specified. Click here for an example. Example: `ciaValue > 4` matches any incident with a varbind value greater than **4**. - **>=** Finds all values greater than or equal to the value specified. Click here for an example. Example: `ciaValue >= 4` matches any incident that contains a varbind with values greater than or equal to **4**. - **between** Finds all traps or events that include a varbind with a value equal to and between the two values specified. Click here for an example. Example: `ciaValue between`  Operator: between Value: 1 4 matches any incident that contains a varbind value equal to or greater than **1** and equal to or less than **4**. **Note**: As shown in the example, each value must be entered on a separate line. - **in** Finds any match to at least one value in a list of values. Click here for an example. Example: `ciaValue in`  Operator: in Value: 4 5 matches any incident with a varbind value of either **4** or **5**. **Note**: As shown in the example, each value must be entered on a separate line. NNMi displays the list of attributes using comma-separated values enclosed in

Payload Filter Editor Components, continued

Attribute	Description
	parentheses, for example (**4, 5**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line. ▪ **is not null** Finds all non-blank values. Click here for an example. Example: `ciaValue is not null` matches any incident with a varbind that contains a value. ▪ **is null** Finds all blank values. Click here for an example. Example: `ciaValue is null` matches any incident with a varbind that does not have a value. ▪ **like** Finds matches using wildcard characters. Click here for more information about using wildcard characters. The period asterisk (.*) characters mean *any number of characters of any type at this location*. The period (.) character means *any single character of any type at this location*. **Note**: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below. Examples: `ciaName like  \Q.1.3.6.1.4.1.9.9\E.*` finds all traps or events that contain varbind names that begin with **.1.3.6.1.4.1.9.9** and (optionally) end with any number of characters. `ciaValue like .*Chicago.*` finds all traps or events that contain a varbind value that includes the string **Chicago**. ▪ **not between** Finds all values except those between the two values specified. Click here for an example. Example: `ciaValue not between 5 8` matches an incident that contains a varbind with the values less than **5** or greater than **8**. ▪ **not in** Finds all values except those included in the list of values. Click here for an example. Example: `ciaValue not in`  Operator: not in Value: 1 2 matches any incident that contains a varbind with values other than **1** and **2**. **Note**: As shown in the example, each value must be entered on a separate line. NNMi displays the list of attributes using comma-separated values enclosed in

Payload Filter Editor Components, continued

Attribute	Description
	parentheses, for example, (**1, 2**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line. ▪ **not like** Finds all that do not have the values specified (using wildcard strings). Click here for an example. The period asterisk (.*) characters mean *any number of characters of any type at this location*. The period (.) character means *any single character of any type at this location*. **Note**: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below. Example: `ciaName not like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that does not begin with **.1.3.6.1.4.1.9.9**. `ciaValue not like .*Chicago.*` finds all traps or events that do not contain a varbind value that includes the string **Chicago**.
Value	The value for which you want NNMi to search. Note the following: ▪ The values you enter are case sensitive. ▪ NNMi displays a variable number of value fields depending on the Operator selected. For example, the `between` Operator causes two value fields to be displayed. ▪ The `between, in` and `not in` operators require that each value be entered on a separate line.

Additional Filters Editor Buttons

Button	Description
Append	Appends the current expression (Attribute, Operator,and Value) to the selected expression already included in the Filter String.
Insert	Inserts the current expression (Attribute, Operator,and Value) in front of the cursor location within the Filter String.
Replace	Replaces the selected expression with the expression displayed in the Attribute, Operator, and Value fields.
AND	Appends, inserts, or replaces the AND Boolean Operator in the selected cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.

Additional Filters Editor Buttons, continued

Button	Description
OR	Appends, Inserts, or replaces the OR Boolean Operator in the current cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
NOT	Can be used in any part of the Filter String to specify that NNMi should exclude interfaces with values that pass the expression that immediately follows the `NOT`. For example, when evaluating the following expression, NNMi includes interfaces with (interface description) `ifDesc` containing **VLAN**, and excludes any Interfaces that have **VLAN10** for the (interface name) `ifName` value: `(ifDesc like VLAN AND NOT (ifName=VLAN10))` **Note**: View the expression displayed under Filter String to see the logic of the expression as it is created .
EXISTS	Used for filters that include Capabilities or Custom Attribute names and values in the Filer String. Indicates that you want NNMi to consider interfaces that do not have any Capabilities or Custom Attributes when evaluating the Filter String. **Note**: If you include Capabilities or Custom Attribute names and values in the Filer String, but do not use **EXISTS** or **NOT EXISTS**, NNMi excludes from its search interfaces that do not include Capabilities or Custom Attributes. For example, when evaluating the following Filter String, NNMi includes interfaces with (interface description) `ifDesc` containing **VLAN**, as well as any Interfaces Custom Attribute Role value is **LAN Connection to Oracle Server**: `(ifDesc like VLAN OR EXISTS((customAttrName=Role AND customAttrValue=LAN Connection to Oracle Server)))` **Note**: View the expression displayed under Filter String to see the logic of the expression as it is created.
NOT EXISTS	Used for filters that include Capabilities or Custom Attribute names and values in the Filer String. Indicates that you want NNMi to consider interfaces that do not have any Capabilities or Custom Attributes when evaluating the Filter String, but exclude the interfaces that match the expression that follows the **NOT EXISTS**. Note: If you include Capabilities or Custom Attribute names and values in the Filer String, but do not use **EXISTS** or **NOT EXISTS**, NNMi excludes from its search interfaces that do not include Capabilities or Custom

Additional Filters Editor Buttons, continued

Button	Description
	Attributes. For example, when evaluating the following expression, NNMi includes interfaces with (interface description) `ifDesc` containing **VLAN**, and excludes any Interfaces that have the Custom Attribute Role and that Role value is **LAN Connection to Oracle Server**: `(ifDesc like VLAN OR NOT EXISTS((customAttrName=Role AND customAttrValue=LAN Connection to Oracle Server))` **Note**: View the expression displayed under Filter String to see the logic of the expression as it is created.
Delete	Deletes the selected expression. **Note**: If the Boolean Operator is selected, the Additional Filters Editor deletes all expressions associated with the Boolean Operator.

Configure Node Settings for an SNMP Trap Incident

NNMi enables you to apply a Suppression, Enrichment, Dampen, Actions, or Diagnostics Selections incident configuration to a Source Node based on the Source Node's participation in a Node Group.

Note: Node Settings override any other Suppression, Enrichment, Dampen, Actions, or Diagnostics Selections configuration settings for this incident, except those configured on the Interface Settings tab.

Tip: See "Create Node Groups" on page 295 for more information about Node Groups.

For information about each Node Settings tab:

For information about each SNMP Traps tab:

To apply an incident configuration to a Source Node based on the Source Node's Node Group:

1. Navigate to the **SNMP Trap Configuration** form:
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Expand the **Incidents** folder.
   c. Select **SNMP Trap Configurations .**
   d. Do one of the following:
      i. To create an incident configuration, click the New icon, and continue.
      ii. To edit an incident configuration, select a row, click the Open icon, and continue.
      iii. To delete an incident configuration, select a row and click the Delete icon.
2. Select the **Node Settings** tab.

3. Do one of the following:
   a. To create a new configuration, click the ✱ New icon.
   b. To edit an existing configuration, select a row, click the Open icon, and continue.
4. Configure the desired Node Settings (see table).
5. Click **Save and Close** to save your changes and return to the previous form.

Node Group Attributes

Name	Description
Node Group	Click the Lookup icon and select Quick Find to select the Node Group you want to use. See "Use the Quick Find Window" on page 41 for more information about using Quick Find.
Ordering	Determines the priority order for those nodes that appear in multiple Node Groups. The lower the number, the higher the priority. For example, **1** is the highest priority. If a node is in multiple Node Groups and more than one of those Node Groups have been specified in an incident configuration, only the incident configuration with the highest priority will be applied to the node.
Enable	Use this attribute to temporarily disable an incident's suppression settings: **Enable** □ = Temporarily disable the selected configuration. **Enable** ☑ = Enable the selected configuration.

Configure Incident Suppression Settings for a Node Group (SNMP Trap Incident)

Note: Node Settings override any other Suppression settings for this incident, except those configured on the Interface Settings tab.

NNMi enables you to suppress a specified incident configuration based on the Source Node's participation in a Node Group. When an incident is suppressed:

- It is not stored in the NNMi database
- It does not appear in an incident view in the NNMi console

You can also suppress the incident configuration based on either of the following:

- The Source Object's participation in an Interface Group. See "Configure Incident Suppression Settings for an Interface Group (SNMP Trap Incident)" on page 804 for more information.
- Incident configuration default settings without specifying a Node or Interface Group. See "Configure Suppression Settings for an SNMP Trap Incident" on page 879 for more information.

Tip: See "Create Node Groups" on page 295 for more information about Node Groups.

For information about each Node Settings tab:

To suppress an incident configuration based on a Node Group:

1. Navigate to the **SNMP Trap Configuration** form:
   1. From the workspace navigation panel, select the **Configuration** workspace.
   2. Expand the **Incidents** folder.
   3. Select **SNMP Trap Configurations** .
   4. Do one of the following:
      1. To create an incident configuration, click the New icon, and continue.
      2. To edit an incident configuration, select a row, click the Open icon, and continue.
      3. To delete an incident configuration, select a row and click the Delete icon.
2. Select the **Node Settings** tab.
3. Do one of the following:
   1. To create a new configuration, click the New icon.
   2. To edit an existing configuration, select a row, and click the Open icon.
4. Make sure you configure the basic Node Setting behavior. See "Configure Node Settings for an SNMP Trap Incident" on page 840 for more information.
5. Select the **Suppression** tab.
6. Configure the desired Suppression behavior (see table).
7. Click **Save and Close** to save your changes and return to the previous form.

Node Settings Suppression Attributes

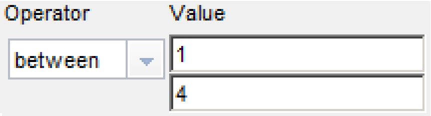
Name	Description
Enable	Use this attribute to temporarily disable an incident's suppression settings: **Enable** □ = Temporarily disable the selected configuration. **Enable** ☑ = Enable the selected configuration.
Payload Filter	The Payload Filter Editor enables you to create expressions that further refine the filters used to select the incidents to be suppressed, enriched, or dampened. Make sure to design any complex Payload Filters offline as a Boolean expression first. This method can help to minimize errors when entering your expressions using the Payload Filter editor. When creating a Payload Filter, note the following: • Payload Filter expressions for the `like` and `not like` operators use the syntax defined for java regular expressions (java.util.regex Pattern class). • You must use a `ciaName` that already exists in the trap or event you are configuring. • Each set of expressions associated with a Boolean Operator is treated as if it were enclosed in parentheses and evaluated together. • View the expression displayed under **Filter String** to see the logic of the expression as it is created.

Node Settings Suppression Attributes , continued

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td></td><td>
• The AND and OR Boolean Operators must contain at least two expressions as shown in the example below.
The following example filters incidents on voltage state:
<pre>
AND
    ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7
    ciaValue = 5
</pre>
NNMi evaluates the expression above as follows:
<pre>
(ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7 AND ciaValue = 5)
</pre>
NNMi finds all incidents with a varbind .1.3.6.1.4.1.9.9.13.1.2.1.7 value of **5**.
• The placement of your cursor and the subsequent text that is selected is important when performing operations using the Payload Filter Editor. For example, you append to, replace, or change the indentation of the expression that is selected.
• The placement of your cursor and the subsequent text that is selected is especially important when adding your Boolean operators.
• You can include more than one varbind in the same Payload Filter expression as shown in the following example:
<pre>
((ciaName like \Q.1.3.6.1.4.1.9.9\E.* AND ciaValue = 25) AND
(ciaName like \Q.1.3.6.1.2.1.2.2.1.1.3\E.* AND ciaValue = 3))
</pre>
In this example, a given trap must meet each of the following criteria:
▪ Contain a varbind whose Object Identifier (OID) matches the regular expression \Q.1.3.6.1.4.1.9.9\E.* and has a value of 25.
▪ Contain a varbind whose OID matches the regular expression \Q.1.3.6.1.2.1.2.2.1.1.3\E.* and has a value of 3.
Payload Filter Editor Components
<table>
<tr><th>Attribute</th><th>Description</th></tr>
<tr><td>Attribute</td><td>The attribute name on which NNMi searches. Filterable attributes include the following:
• ciaName
• ciaValue</td></tr>
<tr><td>Operator</td><td>Valid operators are described below.
• **=** Finds all values equal to the value specified. Click here for an example.
Example: ciaName=.1.3.6.1.4.1.9.9.13.1.2.1.7 matches any</td></tr>
</table>
</td></tr>
</table>

Node Settings Suppression Attributes , continued

Name	Description
	Payload Filter Editor Components, continued

Attribute	Description
	incident that contains a varbind with the name value of **.1.3.6.1.4.1.9.9.13.1.2.1.7**. • **!=** Finds all values not equal to the value specified. Click here for an example. Example: `ciaName! = .1.3.6.1.4.1.9.9.13.1.2.1.7` matches any incident that contains a varbind with a name value other than **1.3.6.1.4.1.9.9.13.1.2.1.7**. • **<** Finds all values less than the value specified. Click here for an example. Example: `ciaValue < 6` matches any incident that contains a varbind with a value less than **6**. • **<=** Finds all values less than or equal to the value specified. Click here for an example. Example: `ciaValue <= 6` matches any incident that contains a varbind with a value less than or equal to **6**. • **>** Finds all values greater than the value specified. Click here for an example. Example: `ciaValue > 4` matches any incident that contains a varbind with a value greater than **4**. • **>=** Finds all values greater than or equal to the value specified. Click here for an example. Example: `ciaValue >= 4` matches any incident that contains a varbind with values greater than or equal to **4**. • **between** Finds all values equal to and between the two values specified. Click here for an example. Example: `ciaValue between`  Operator: between Value: 1 4 matches any incident that contains a varbind value equal to or greater than **1** and equal to or less than **4**. **Note**: As shown in the example, each value must be entered on a separate line. • **in** Finds any match to at least one value in a list of values. Click here for an example.

Node Settings Suppression Attributes , continued

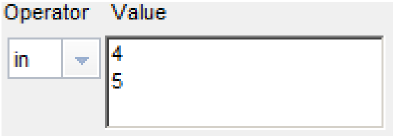
<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td></td><td>

Payload Filter Editor Components, continued

<table>
<tr><th>Attribute</th><th>Description</th></tr>
<tr><td></td><td>

Example:

`ciaValue in`



Operator: in
Value:
4
5

matches any incident with a varbind value of either **4** or **5**.

Note: As shown in the example, each value must be entered on a separate line.

NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example (**4, 5**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line.

- **is not null** Finds all non-blank values. Click here for an example.

  Example: `ciaValue is not null` matches any incident with a varbind that contains a value.

- **is null** Finds all blank values. Click here for an example.

  Example: `ciaValue is null` matches any incident with a varbind that does not contain a value.

- **like** Finds matches using the syntax defined for java regular expressions. See the Pattern (Java Platform SE6) API documentation at : `http://download.oracle.com/javase/6/docs/api/java/util/regex/Pattern.html` for more information. Click here for more information.

  The period asterisk (.*) characters mean *any number of characters of any type at this location*.

  The period (.) character means *any single character of any type at this location*.

  Note: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below.

  Example:

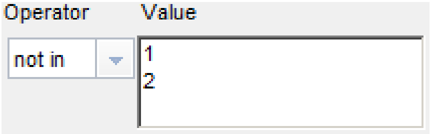
  `ciaName like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident

</td></tr>
</table>

</td></tr>
</table>

Node Settings Suppression Attributes , continued

Name	Description
	Payload Filter Editor Components, continued

Attribute	Description
	that contains a varbind name value that begins with 1.3.6.1.4.1.9.9 and (optionally) ends with any number of characters. `ciaValue like .*Chicago.*` finds all traps or events that contain a varbind value that includes the string **Chicago**. • **not between** Finds all values except those between the two values specified. Click here for an example. Example: `ciaValue not between 5 8` matches an incident that contains a varbind with the values less than **5** or greater than **8** . • **not in** Finds all values except those included in the list of values. Click here for an example. Example: `ciaValue not in`  Operator: not in Value: 1 2 matches any incident that contains a varbind with values other than **1** and **2**. **Note**: As shown in the example, each value must be entered on a separate line. NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example, (**1, 2**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line. • **not like** Finds all that do not have the values specified using the syntax defined for Java regular expressions. See the Pattern (Java Platform SE6) API documentation at: `http://download.oracle.com/javase/6/docs/api/java/util/regex/Pattern.html` for more information. Click here for an example. The period asterisk (.*) characters mean *any number of characters of any type at this location*. The period (.) character means *any single character of any type at this location*.

Node Settings Suppression Attributes , continued

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td></td><td>

Payload Filter Editor Components, continued

<table>
<tr><th>Attribute</th><th>Description</th></tr>
<tr><td></td><td>

Note: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below.

Example:

`ciaName not like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that does not begin with **.1.3.6.1.4.1.9.9** and (optionally) ends with any number of characters.

`ciaValue not like .*Chicago.*` finds all traps or events that do not contain a varbind value that includes the string **Chicago**.

</td></tr>
<tr><td>Value</td><td>

The value for which you want NNMi to search.

Note the following:

- The values you enter are case sensitive.
- NNMi displays a variable number of value fields depending on the Operator selected. For example, the `between` Operator causes two value fields to be displayed.
- The `between, in` and `not in` operators require that each value be entered on a separate line.

</td></tr>
</table>

Payload Filter Editor Buttons

<table>
<tr><th>Button</th><th>Description</th></tr>
<tr><td>Append</td><td>Appends the current expression (Attribute, Operator,and Value) to the selected expression already included in the filter string.</td></tr>
<tr><td>Insert</td><td>Inserts the current expression (Attribute, Operator,and Value) in front of the cursor location within the Filter String.</td></tr>
<tr><td>Replace</td><td>Replaces the selected expression with the expression displayed in the Attribute, Operator, and Value fields.</td></tr>
<tr><td>AND</td><td>

Inserts the AND Boolean Operator in the selected cursor location.

Note: View the expression displayed under **Filter String** to see the logic of the expression as it is created.

</td></tr>
<tr><td>OR</td><td>Inserts the OR Boolean Operator in the current cursor location.</td></tr>
</table>

</td></tr>
</table>

Node Settings Suppression Attributes , continued

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td></td><td>
<b>Payload Filter Editor Buttons, continued</b>
<table>
<tr><th>Button</th><th>Description</th></tr>
<tr><td></td><td><b>Note</b>: View the expression displayed under <b>Filter String</b> to see the logic of the expression as it is created.</td></tr>
<tr><td>NOT</td><td>Can be used in any part of the Filter String to specify that NNMi should exclude interfaces with values that pass the expression that immediately follows the <code>NOT</code>.
For example, when evaluating the following expression, NNMi includes interfaces with (interface description) <code>ifDesc</code> containing <b>VLAN</b>, and excludes any Interfaces that have <b>VLAN10</b> for the (interface name) <code>ifName</code> value:
<code>(ifDesc like VLAN AND NOT (ifName=VLAN10))</code>
<b>Note</b>: View the expression displayed under Filter String to see the logic of the expression as it is created .</td></tr>
<tr><td>EXISTS</td><td>Used for filters that include Capabilities or Custom Attribute names and values in the Filer String.
Indicates that you want NNMi to consider interfaces that do not have any Capabilities or Custom Attributes when evaluating the Filter String.
<b>Note</b>: If you include Capabilities or Custom Attribute names and values in the Filer String, but do not use <b>EXISTS</b> or <b>NOT EXISTS</b>, NNMi excludes from its search interfaces that do not include Capabilities or Custom Attributes.
For example, when evaluating the following Filter String, NNMi includes interfaces with (interface description) <code>ifDesc</code> containing <b>VLAN</b>, as well as any Interfaces Custom Attribute Role value is <b>LAN Connection to Oracle Server</b>:
<code>(ifDesc like VLAN OR EXISTS((customAttrName=Role AND customAttrValue=LAN Connection to Oracle Server)))</code>
<b>Note</b>: View the expression displayed under Filter String to see the logic of the expression as it is created.</td></tr>
<tr><td>NOT EXISTS</td><td>Used for filters that include Capabilities or Custom Attribute names and values in the Filer String. Indicates that you want NNMi to consider interfaces that do not have any Capabilities or Custom Attributes when evaluating the Filter String, but exclude the interfaces that match the expression that follows the <b>NOT EXISTS</b>.</td></tr>
</table>
</td></tr>
</table>

Node Settings Suppression Attributes , continued

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td></td><td>
<b>Payload Filter Editor Buttons, continued</b>
<table>
<tr><th>Button</th><th>Description</th></tr>
<tr><td></td><td><b>Note</b>: If you include Capabilities or Custom Attribute names and values in the Filer String, but do not use <b>EXISTS</b> or <b>NOT EXISTS</b>, NNMi excludes from its search interfaces that do not include Capabilities or Custom Attributes.
For example, when evaluating the following expression, NNMi includes interfaces with (interface description) <code>ifDesc</code> containing <b>VLAN</b>, and excludes any Interfaces that have the Custom Attribute Role and that Role value is <b>LAN Connection to Oracle Server</b>:
<code>(ifDesc like VLAN OR NOT EXISTS ((customAttrName=Role AND customAttrValue=LAN Connection to Oracle Server)))</code>
<b>Note</b>: View the expression displayed under Filter String to see the logic of the expression as it is created.</td></tr>
<tr><td>Delete</td><td>Deletes the selected expression.
<b>Note</b>: If the Boolean Operator is selected, the Payload Filter Editor deletes all expressions associated with the Boolean Operator.</td></tr>
</table>
</td></tr>
</table>

Configure Incident Enrichment Settings for a Node Group (SNMP Trap Incident)

Note: Node Settings override any other Enrichment configuration for this incident, except those configured on the Interface Settings tab.

NNMi enables you to enhanced a specified incident configuration based on the Source Node's participation in a Node Group. The types of items you can fine tune and enhance for a selected incident configuration, include:

- Category
- Family
- Severity
- Priority
- Correlation Nature
- Message
- Assigned To

You can also enrich the incident configuration based on either of the following:

- The Source Object's participation in an Interface Group. See "Configure Incident Enrichment Settings for an Interface Group (SNMP Trap Incident)" on page 812 for more information.
- Incident configuration default settings without specifying a Node or Interface Group. See "Configure Enrichment Settings for an SNMP Trap Incident" on page 887 for more information.

For information about each Node Settings tab:

For information about each Enrichment tab:

To configure enrichment settings for a Node Group:

1. Navigate to the **SNMP Trap Configuration** form:
    a. From the workspace navigation panel, select the **Configuration** workspace.
    b. Expand the **Incidents** folder.
    c. Select **SNMP Trap Configurations** .
    d. Do one of the following:
        i. To create an incident configuration, click the New icon, and continue.
        ii. To edit an incident configuration, select a row, click the Open icon, and continue.
        iii. To delete an incident configuration, select a row and click the Delete icon.
2. Select the **Node Settings** tab.
3. Do one of the following:
    a. To create a new configuration, click the New icon.
    b. To edit an existing configuration, select a row, click the Open icon, and continue.
4. Make sure the basic Node Setting behavior is configured. See "Configure Node Settings for an SNMP Trap Incident" on page 840for more information.
5. Select the **Enrichment** tab.
6. Do one of the following:
    a. To create an Enrichment configuration, click the New icon and continue.
    b. To edit an Enrichment configuration, select a row, click the Open icon, and continue.
    c. To delete an Enrichment configuration, select a row and click the Delete icon.
7. Configure the desired Enrichment behavior (see the "Enrich Configuration Attributes" table)
8. Click **Save and Close** to save your changes and return to the previous form.

Node Settings Enrich Configuration Attributes

Name	Description
Category	Use the Category attribute to customize the category for this incident configuration. Possible values include:

Node Settings Enrich Configuration Attributes , continued

Name	Description
	• Accounting • Application Status • Configuration • Fault • Performance • Security • Status See "Specify Category and Family Attribute Values for Organizing Your Incidents (SNMP Trap Incident)" on page 790 for more information.
Family	Use the Family attribute to customize the Family for this incident configuration. Select from the drop-down list or create a new value. For example, some of the values provided by NNMi include: • Address • Aggregated Port (Interfaces using **Link Aggregation**[1] protocol. See Interface Form: Link Aggregation tab.) • Card • Connection • Correlation • Interface • Node
Severity	The incident Severity represents the seriousness calculated for the incident. Use the Severity attribute to specify the severity that should be assigned to the incident you are configuring. Possible values are described below: **Normal** - Indicates there are no known problems related to the associated object. This Severity is meant to be informational. Generally, no action is needed for these incidents. **Warning** - Indicates there might be a problem related to the associated object. **Minor** - Indicates NNMi has detected problems related to the associated object

[1] Protocols used on Switches to configure multiple Interfaces (Aggregation Member Interfaces) to function as if they were one (an Aggregator Interface). When two Aggregator Interfaces establish a connection, that connection is an Aggregator Layer 2 Connection. The Aggregator Layer 2 Connection appears on Layer 2 Neighbor View maps as a thick line with an Interface icon at each end (representing the Aggregator Interface). When you double-click the thick line, it converts into multiple thin lines representing the participating Aggregation Member Layer 2 Connections with their Aggregation Member Interfaces at each end of the lines.

Node Settings Enrich Configuration Attributes , continued

Name	Description
	that require further investigation. **Major** - Indicates NNMi has detected problems related to the associated object to be resolved before they become critical. **Critical** - Indicates NNMi has detected problems related to the associated object that require immediate attention.
Priority	Used to communicate the urgency of resolving the selected incident. You control this value. NNMi sets this value to null by default. The lower the number the higher the priority. Possible values are: 5 **None** 4 **Low** 3 **Medium** 2 **High** 1 **Top** **Note**: The icons are displayed only in table views.
Correlation Nature	Use the Correlation Nature to customize the Correlation Nature for this incident configuration. Possible values include: • Info • None • Root Cause • Secondary Root Cause • Symptom • Stream Correlation • Service Impact • Dedup Stream Correlation • Rate Stream Correlation See Incident Form: General Tab for more information.
Message Format	When configuring an incident, specify how the incident message appears in the incident view. The string you specify in the Message Format attribute is visible in an incident view. **Note**: The incident message limit is 1024 characters. If you exceed this limit, NNMi truncates the value starting from the right.

Node Settings Enrich Configuration Attributes , continued

Name	Description
	You can use any combination of default and custom attributes: "Valid Parameters for Configuring Incident Messages (SNMP Trap Incident)" on page 795 "Include Custom Incident Attributes in Your Message Format (SNMP Trap Incident)" on page 801
Assigned To	Use to specify the owner of any incident generated for this incident configuration. Click the Lookup icon and select Quick Find to select a valid user name. **Note**: You can also begin to type a valid user name and use the auto-complete feature to select the user name of interest.
Description	Use the Description attribute to provide additional information that you want to note about the current enhancement configuration. This description applies only to the enhancement configuration and does not appear when NNMi displays any associated incident. Type a maximum of 1024 characters. Alpha-numeric and special characters (~ ! @ # $ % ^ & * ( ) _+ -) are permitted.

Configure Custom Incident Attributes to Enrich an Incident Configuration (Node Settings) (SNMP Trap Incidents)

The Custom Incident Attributes (CIAs) tab enables you to specify additional CIAs to be saved with an incoming incident. The persisted data might then be used as an argument to an action for the incident.

When creating a CIA for an incident configuration, you can specify any of the following values:

- Custom Attribute on the source node
- Custom Attribute on the interface (source object)

You also specify the Custom Incident Attribute name that will store this information.

For information about each Enrichment tab:

To create a Custom Incident Attribute to enrich an incident configuration:

1. Navigate to the **SNMP Trap Configuration** form:
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Expand the **Incidents** folder.
   c. Select **SNMP Trap Configurations** .
   d. Do one of the following:

        i. To create an incident configuration, click the New icon, and continue.
        ii. To edit an incident configuration, select a row, click the Open icon, and continue.
        iii. To delete an incident configuration, select a row and click the Delete icon.

2. Select the **Node Settings** tab.
3. Do one of the following:
    a. To create a new configuration, click the New icon.
    b. To edit an existing configuration, select a row, click the Open icon, and continue.
    c. To delete an existing configuration, select a row and click the Delete icon.
4. Make sure you configure the basic Node Setting behavior. See "Configure Node Settings for an SNMP Trap Incident" on page 840for more information.
5. Select the **Enrichment** tab.
6. Do one of the following:
    a. To create an Enrichment configuration, click the New icon, and continue.
    b. To edit an Enrichment configuration,select a row, click the Open icon, and continue.
    c. To delete an Enrichment configuration, select a row and click the Delete icon.
7. Make sure the Enrichment settings are configured. See "Configure Incident Enrichment Settings for a Node Group (SNMP Trap Incident)" on page 849 for more information.
8. Navigate to the **Custom Incident Attributes** tab.
9. Do one of the following:
    a. To create a Custom Incident Attribute, click the New icon, and continue.
    b. To edit a Custom Incident Attribute, select a row, click the Open icon, and continue.
    c. To delete a Custom Incident Attribute, select a row and click the Delete icon.
10. Configure the Custom Incident Attribute (see the "Custom Incident Attribute" table).
11. Click **Save and Close** to save your changes and return to the previous form.

Custom Incident Attribute

Name	Description
Type	Specifies whether you are using a Custom Attribute on a node or a Custom Attribute on an interface. Possible values are: • Node Custom Attribute • Interface Custom Attribute
Custom Attribute Name	Used to determine the value to be assigned to the Custom Incident Attribute you are configuring. Enter either of the following: • Name of the Custom Attribute on the source node

Custom Incident Attribute , continued

Name	Description
	• Name of the Custom Attribute on the interface (source object)
Custom Incident Attribute Name	Name used to identify the Custom Incident Attribute you are configuring. The name limit is 255 characters. Alpha-numeric characters are permitted. No spaces or special characters (~ ! @ # $ % ^ & * ( ) _+ -) are permitted.

Configure a Payload Filter to Enrich an Incident Configuration (Node Settings) (SNMP Trap Incidents)

The Payload Filter Editor enables you to create expressions that further refine the filters used to select the incidents to be enriched. Make sure to design any complex Payload Filters offline as a Boolean expression first. This method can help to minimize errors when entering your expressions using the Payload Filter editor.

To create a Payload Filter expression:

1. Navigate to the **SNMP Trap Configuration** form:
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Expand the **Incidents** folder.
   c. Select **SNMP Trap Configurations** .
   d. Do one of the following:
      i. To create an incident configuration, click the ✳ New icon, and continue.
      ii. To edit an incident configuration, select a row, click the 📂 Open icon, and continue.
      iii. To delete an incident configuration, select a row and click the ✖ Delete icon.
2. Select the **Node Settings** tab.
3. Do one of the following:
   a. To create a new configuration, click the ✳ New icon.
   b. To edit an existing configuration, select a row, click the 📂 Open icon, and continue.
   c. To delete an existing configuration, select a row and click the ✖ Delete icon.
4. Make sure the basic Node Setting behavior is configured. See "Configure Node Settings for an SNMP Trap Incident" on page 840for more information.
5. Select the **Enrichment** tab.
6. Do one of the following:
   a. To create an Enrichment configuration, click the ✳ New icon, and continue.

b. To edit an Enrichment configuration, select a row, click the Open icon, and continue.

c. To delete an Enrichment configuration, select a row and click the ✕ Delete icon.

7. Make sure the Enrichment settings are configured. See "Configure Incident Enrichment Settings for a Node Group (SNMP Trap Incident)" on page 849 for more information.

8. Select the **Payload Filter** tab.

9. Define your Payload Filter (see table). Also see "Guidelines for Creating a Payload Filter".

   a. Plan out the logic needed for your Filter String.

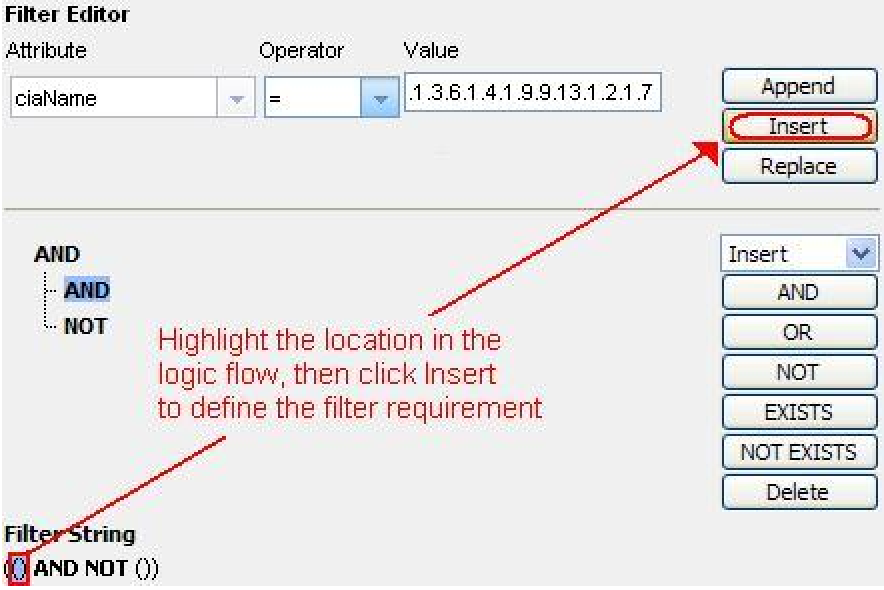
   b. Use the buttons on the bottom half of the Additional Filters Editor to establish the logic structure.

   For example, to establish the following structure, select **Insert**, then click **AND**, then **NOT**, and then **AND** a second time:

   ```
   (( ) AND NOT ( ))
   ```

   c. Now place your cursor in a location within the displayed Filter String, and use the top half of the filter editor to define the parameters of the highlighted filter requirement.

   For example, select a set of parentheses and use the Insert button to specify the filter requirement within those parentheses:



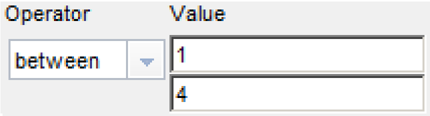
10. Click **Save and Close**.

11. Click **Save and Close** to save your changes and return to the previous form.

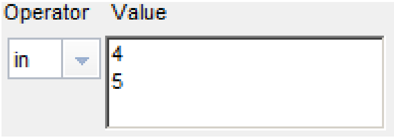
Payload Filter Editor Components

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: • ciaName

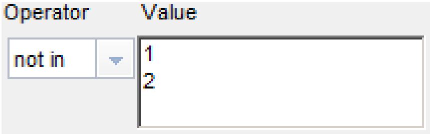
Payload Filter Editor Components, continued

Attribute	Description
	• ciaValue
Operator	Valid operators are described below. • **=** Finds all values equal to the value specified. Click here for an example. Example: `ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7` matches any incident that contains a varbind with the name value **.1.3.6.1.4.1.9.9.13.1.2.1.7**. • **!=** Finds all values not equal to the value specified. Click here for an example. Example: `ciaName != .1.3.6.1.4.1.9.9.13.1.2.1.7` matches any incident that contains a varbind with the name value other than **1.3.6.1.4.1.9.9.13.1.2.1.7**. • **<** Finds all values less than the value specified. Click here for an example. Example: `ciaValue < 6` matches any incident with a varbind value less than **6**. • **<=** Finds all values less than or equal to the value specified. Click here for an example. Example: `ciaValue <= 6` matches any incident with a varbind value less than or equal to **6**. • **>** Finds all values greater than the value specified. Click here for an example. Example: `ciaValue > 4` matches any incident with a varbind value greater than **4**. • **>=** Finds all values greater than or equal to the value specified. Click here for an example. Example: `ciaValue >= 4` matches any incident that contains a varbind with values greater than or equal to **4**. • **between** Finds all traps or events that include a varbind with a value equal to and between the two values specified. Click here for an example. Example: `ciaValue between`  Operator: between Value: 1 4 matches any incident that contains a varbind value equal to or greater than **1** and equal to or less than **4**. **Note**: As shown in the example, each value must be entered on a separate line. • **in** Finds any match to at least one value in a list of values. Click here for an example. Example: `ciaValue in`

Payload Filter Editor Components, continued

Attribute	Description
	 Operator Value in 4 5 matches any incident with a varbind value of either **4** or **5**. **Note**: As shown in the example, each value must be entered on a separate line. NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example (**4, 5**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line. • **is not null** Finds all non-blank values. Click here for an example. Example: `ciaValue is not null` matches any incident with a varbind that contains a value. • **is null** Finds all blank values. Click here for an example. Example: `ciaValue is null` matches any incident with a varbind that does not have a value. • **like** Finds matches using wildcard characters. Click here for more information about using wildcard characters. The period asterisk (.*) characters mean *any number of characters of any type at this location*. The period (.) character means *any single character of any type at this location*. **Note**: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below. Examples: `ciaName like  \Q.1.3.6.1.4.1.9.9\E.*` finds all traps or events that contain varbind names that begin with **.1.3.6.1.4.1.9.9** and (optionally) end with any number of characters. `ciaValue like .*Chicago.*` finds all traps or events that contain a varbind value that includes the string **Chicago**. • **not between** Finds all values except those between the two values specified. Click here for an example. Example: `ciaValue not between 5 8` matches an incident that contains a varbind with the values less than **5** or greater than **8**. • **not in** Finds all values except those included in the list of values. Click here for an example. Example: `ciaValue not in`

Payload Filter Editor Components, continued

Attribute	Description
	 1 2 matches any incident that contains a varbind with values other than **1** and **2**. **Note**: As shown in the example, each value must be entered on a separate line. NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example, (**1, 2**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line. - **not like** Finds all that do not have the values specified (using wildcard strings). Click here for an example. The period asterisk (.*) characters mean *any number of characters of any type at this location*. The period (.) character means *any single character of any type at this location*. **Note**: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below. Example: `ciaName not like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that does not begin with **.1.3.6.1.4.1.9.9**. `ciaValue not like .*Chicago.*` finds all traps or events that do not contain a varbind value that includes the string **Chicago**.
Value	The value for which you want NNMi to search. Note the following: - The values you enter are case sensitive. - NNMi displays a variable number of value fields depending on the Operator selected. For example, the `between` Operator causes two value fields to be displayed. - The `between, in` and `not in` operators require that each value be entered on a separate line.

Additional Filters Editor Buttons

Button	Description
Append	Appends the current expression (Attribute, Operator,and Value) to the selected expression already included in the Filter String.
Insert	Inserts the current expression (Attribute, Operator,and Value) in front of the cursor location within the Filter String.

Additional Filters Editor Buttons, continued

Button	Description
Replace	Replaces the selected expression with the expression displayed in the Attribute, Operator, and Value fields.
AND	Appends, inserts, or replaces the AND Boolean Operator in the selected cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
OR	Appends, Inserts, or replaces the OR Boolean Operator in the current cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
NOT	Can be used in any part of the Filter String to specify that NNMi should exclude interfaces with values that pass the expression that immediately follows the `NOT`. For example, when evaluating the following expression, NNMi includes interfaces with (interface description) `ifDesc` containing **VLAN**, and excludes any Interfaces that have **VLAN10** for the (interface name) `ifName` value: `(ifDesc like VLAN AND NOT (ifName=VLAN10))` **Note**: View the expression displayed under Filter String to see the logic of the expression as it is created .
EXISTS	Used for filters that include Capabilities or Custom Attribute names and values in the Filer String. Indicates that you want NNMi to consider interfaces that do not have any Capabilities or Custom Attributes when evaluating the Filter String. **Note**: If you include Capabilities or Custom Attribute names and values in the Filer String, but do not use **EXISTS** or **NOT EXISTS**, NNMi excludes from its search interfaces that do not include Capabilities or Custom Attributes. For example, when evaluating the following Filter String, NNMi includes interfaces with (interface description) `ifDesc` containing **VLAN**, as well as any Interfaces Custom Attribute Role value is **LAN Connection to Oracle Server**: `(ifDesc like VLAN OR EXISTS((customAttrName=Role AND customAttrValue=LAN Connection to Oracle Server)))` **Note**: View the expression displayed under Filter String to see the logic of the expression as it is created.
NOT EXISTS	Used for filters that include Capabilities or Custom Attribute names and values in the Filer String. Indicates that you want NNMi to consider interfaces that do not have any Capabilities or Custom Attributes when evaluating the Filter String, but exclude the interfaces that match the expression that follows the **NOT EXISTS**.

Additional Filters Editor Buttons, continued

Button	Description
	Note: If you include Capabilities or Custom Attribute names and values in the Filer String, but do not use **EXISTS** or **NOT EXISTS**, NNMi excludes from its search interfaces that do not include Capabilities or Custom Attributes. For example, when evaluating the following expression, NNMi includes interfaces with (interface description) `ifDesc` containing **VLAN**, and excludes any Interfaces that have the Custom Attribute Role and that Role value is **LAN Connection to Oracle Server**: `(ifDesc like VLAN OR NOT EXISTS((customAttrName=Role AND customAttrValue=LAN Connection to Oracle Server)))` **Note**: View the expression displayed under Filter String to see the logic of the expression as it is created.
Delete	Deletes the selected expression. **Note**: If the Boolean Operator is selected, the Additional Filters Editor deletes all expressions associated with the Boolean Operator.

Configure Incident Dampening Settings for a Node Group (SNMP Trap Incident)

Note: Node Settings override any other Dampening settings for this incident, except those configured on the Interface Settings tab.

NNMi enables you to delay the following for an incident configuration based on the Source Node's participation in a Node Group:

- Execution of Incident Actions
- Execution of Diagnostics (*HP Network Node Manager iSPI Network Engineering Toolset Software \ NNM iSPI NET*)
- Appearance within Incident views in the NNMi Console

You can also configure Dampening settings based on either of the following:

- The Source Object's participation in an Interface Group. See "Configure Incident Dampening Settings for an Interface Group (SNMP Trap Incident)" on page 824 for more information.
- Incident configuration default settings without specifying a Node or Interface Group. See "Configure Dampening Settings for an SNMP Trap Incident" on page 892 for more information.

Tip: See "Create Node Groups" on page 295 for more information about Node Groups.

For information about each Node Settings tab:

After the Dampen Interval has passed, NNMi changes the Lifecycle State to REGISTERED.

See About the Incident Lifecycle for more information about Lifecycle State.

To configure Dampening settings based on a Node Group:

1. Navigate to the **SNMP Trap Configuration** form:
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Expand the **Incidents** folder.
   c. Select **SNMP Trap Configurations**.
   d. Do one of the following:
      i. To create an incident configuration, click the New icon, and continue.
      ii. To edit an incident configuration, select a row, click the Open icon, and continue.
      iii. To delete an incident configuration, select a row and click the Delete icon.
2. Select the **Node Settings** tab.
3. Do one of the following:
   a. To create a new configuration, click the New icon.
   b. To edit an existing configuration, select a row, click the Open icon, and continue.
   c. To delete an existing configuration, select a row and click the Delete icon.
4. Make sure you configure the basic Node Setting behavior. See "Configure Node Settings for an SNMP Trap Incident" on page 840 for more information.
5. Select the **Dampen** tab.
6. Configure the desired Dampen behavior (see table).
7. Click **Save and Close** to save your changes and return to the previous form.

Node Settings Dampen Attributes

Name	Description
Enable	Use this attribute to temporarily disable an incident's Dampening settings: **Enable** □ = Temporarily disable the selected configuration. **Enable** ☑ = Enable the selected configuration.
Hour	Specifies the number of hours to be used for the dampen interval.
Minutes	Specifies the number of minutes to be used for the dampen interval. **Note:** The maximum dampen interval is 60 minutes.
Seconds	Specifies the number of seconds to be used for the dampen interval.
Payload Filter	The Payload Filter Editor enables you to create expressions that further refine the filters used to select the incidents to be suppressed, enriched, or dampened. Make sure to design any complex Payload Filters offline as a Boolean expression first. This method can help to minimize errors when entering your expressions using the Payload Filter editor.

Node Settings Dampen Attributes , continued

Name	Description

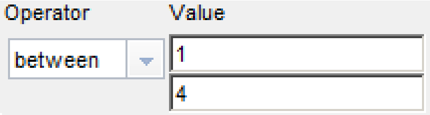
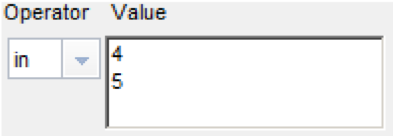
	When creating a Payload Filter, note the following:
	• Payload Filter expressions for the `like` and `not like` operators use the syntax defined for java regular expressions (java.util.regex Pattern class).
	• You must use a `ciaName` that already exists in the trap or event you are configuring.
	• Each set of expressions associated with a Boolean Operator is treated as if it were enclosed in parentheses and evaluated together.
	• View the expression displayed under **Filter String** to see the logic of the expression as it is created.
	• The `AND` and `OR` Boolean Operators must contain at least two expressions as shown in the example below.
	The following example filters incidents on voltage state:
	`AND`
	`    ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7`
	`    ciaValue = 5`
	NNMi evaluates the expression above as follows:
	`(ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7 AND ciaValue = 5)`
	NNMi finds all incidents with a varbind `.1.3.6.1.4.1.9.9.13.1.2.1.7` value of **5**.
	• The placement of your cursor and the subsequent text that is selected is important when performing operations using the Payload Filter Editor. For example, you append to, replace, or change the indentation of the expression that is selected.
	• The placement of your cursor and the subsequent text that is selected is especially important when adding your Boolean operators.
	• You can include more than one varbind in the same Payload Filter expression as shown in the following example:
	`((ciaName like \Q.1.3.6.1.4.1.9.9\E.* AND ciaValue = 25) AND (ciaName like \Q.1.3.6.1.2.1.2.2.1.1.3\E.* AND ciaValue = 3))`
	In this example, a given trap must meet each of the following criteria:
	▪ Contain a varbind whose Object Identifier (OID) matches the regular expression `\Q.1.3.6.1.4.1.9.9\E.*` and has a value of `25`.
	▪ Contain a varbind whose OID matches the regular expression `\Q.1.3.6.1.2.1.2.2.1.1.3\E.*` and has a value of `3`.

Node Settings Dampen Attributes , continued

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td></td><td>
<b>Payload Filter Editor Components</b>
<table>
<tr><th>Attribute</th><th>Description</th></tr>
<tr><td>Attribute</td><td>The attribute name on which NNMi searches. Filterable attributes include the following:
<ul><li>ciaName</li><li>ciaValue</li></ul></td></tr>
<tr><td>Operator</td><td>Valid operators are described below.
<ul>
<li><b>=</b> Finds all values equal to the value specified. Click here for an example.
Example: <code>ciaName=.1.3.6.1.4.1.9.9.13.1.2.1.7</code> matches any incident that contains a varbind with the name value of <b>.1.3.6.1.4.1.9.9.13.1.2.1.7</b>.</li>
<li><b>!=</b> Finds all values not equal to the value specified. Click here for an example.
Example: <code>ciaName! = .1.3.6.1.4.1.9.9.13.1.2.1.7</code> matches any incident that contains a varbind with a name value other than <b>1.3.6.1.4.1.9.9.13.1.2.1.7</b>.</li>
<li><b>&lt;</b> Finds all values less than the value specified. Click here for an example.
Example: <code>ciaValue &lt; 6</code> matches any incident that contains a varbind with a value less than <b>6</b>.</li>
<li><b>&lt;=</b> Finds all values less than or equal to the value specified. Click here for an example.
Example: <code>ciaValue &lt;= 6</code> matches any incident that contains a varbind with a value less than or equal to <b>6</b>.</li>
<li><b>&gt;</b> Finds all values greater than the value specified. Click here for an example.
Example: <code>ciaValue &gt; 4</code> matches any incident that contains a varbind with a value greater than <b>4</b>.</li>
<li><b>&gt;=</b> Finds all values greater than or equal to the value specified. Click here for an example.
Example: <code>ciaValue &gt;= 4</code> matches any incident that contains a varbind with values greater than or equal to <b>4</b>.</li>
<li><b>between</b> Finds all values equal to and between the two values specified. Click here for an example.
Example: <code>ciaValue between</code></li>
</ul></td></tr>
</table>
</td></tr>
</table>

Node Settings Dampen Attributes , continued

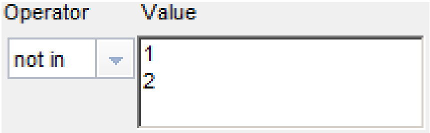
Name	Description
	Payload Filter Editor Components, continued

Attribute	Description
	 Operator Value between 1 4 matches any incident that contains a varbind value equal to or greater than **1** and equal to or less than **4**. **Note**: As shown in the example, each value must be entered on a separate line. - **in** Finds any match to at least one value in a list of values. Click here for an example. Example: `ciaValue in`  Operator Value in 4 5 matches any incident with a varbind value of either **4** or **5**. **Note**: As shown in the example, each value must be entered on a separate line. NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example (**4, 5**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line. - **is not null** Finds all non-blank values. Click here for an example. Example: `ciaValue is not null` matches any incident with a varbind that contains a value. - **is null** Finds all blank values. Click here for an example. Example: `ciaValue is null` matches any incident with a varbind that does not contain a value. - **like** Finds matches using the syntax defined for java regular expressions. See the Pattern (Java Platform SE6) API documentation at : `http://download.oracle.com/javase/6/docs/api/java/util/regex/Pattern.html` for more information. Click here for more information. The period asterisk (.*) characters mean *any number of characters of any*

Node Settings Dampen Attributes , continued

Name	Description

Payload Filter Editor Components, continued

Attribute	Description
	type at this location. The period (.) character means *any single character of any type at this location*. **Note**: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below. Example: `ciaName like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that begins with 1.3.6.1.4.1.9.9 and (optionally) ends with any number of characters. `ciaValue like .*Chicago.*` finds all traps or events that contain a varbind value that includes the string **Chicago**. • **not between** Finds all values except those between the two values specified. Click here for an example. Example: `ciaValue not between 5 8` matches an incident that contains a varbind with the values less than **5** or greater than **8** . • **not in** Finds all values except those included in the list of values. Click here for an example. Example: `ciaValue not in`  Operator: not in Value: 1 2 matches any incident that contains a varbind with values other than **1** and **2**. **Note**: As shown in the example, each value must be entered on a separate line. NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example, (**1, 2**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line. • **not like** Finds all that do not have the values specified using the syntax defined for Java regular expressions. See the Pattern (Java Platform SE6)

Node Settings Dampen Attributes , continued

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td></td><td>

Payload Filter Editor Components, continued

<table>
<tr><th>Attribute</th><th>Description</th></tr>
<tr><td></td><td>API documentation at: `http://download.oracle.com/javase/6/docs/api/java/util/regex/Pattern.html` for more information. Click here for an example.
The period asterisk (.*) characters mean *any number of characters of any type at this location*.
The period (.) character means *any single character of any type at this location*.
Note: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below.
Example:
`ciaName not like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that does not begin with **.1.3.6.1.4.1.9.9** and (optionally) ends with any number of characters.
`ciaValue not like .*Chicago.*` finds all traps or events that do not contain a varbind value that includes the string **Chicago**.</td></tr>
<tr><td>Value</td><td>The value for which you want NNMi to search.
Note the following:
• The values you enter are case sensitive.
• NNMi displays a variable number of value fields depending on the Operator selected. For example, the `between` Operator causes two value fields to be displayed.
• The `between, in` and `not in` operators require that each value be entered on a separate line.</td></tr>
</table>

Payload Filter Editor Buttons

<table>
<tr><th>Button</th><th>Description</th></tr>
<tr><td>Append</td><td>Appends the current expression (Attribute, Operator,and Value) to the selected expression already included in the filter string.</td></tr>
<tr><td>Insert</td><td>Inserts the current expression (Attribute, Operator,and Value) in front of the cursor location within the Filter String.</td></tr>
<tr><td>Replace</td><td>Replaces the selected expression with the expression displayed in</td></tr>
</table>

</td></tr>
</table>

Node Settings Dampen Attributes , continued

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td></td><td>

Payload Filter Editor Buttons, continued

<table>
<tr><th>Button</th><th>Description</th></tr>
<tr><td></td><td>the Attribute, Operator, and Value fields.</td></tr>
<tr><td>AND</td><td>Inserts the AND Boolean Operator in the selected cursor location.
<b>Note</b>: View the expression displayed under <b>Filter String</b> to see the logic of the expression as it is created.</td></tr>
<tr><td>OR</td><td>Inserts the OR Boolean Operator in the current cursor location.
<b>Note</b>: View the expression displayed under <b>Filter String</b> to see the logic of the expression as it is created.</td></tr>
<tr><td>NOT</td><td>Can be used in any part of the Filter String to specify that NNMi should exclude interfaces with values that pass the expression that immediately follows the <code>NOT</code>.
For example, when evaluating the following expression, NNMi includes interfaces with (interface description) <code>ifDesc</code> containing <b>VLAN</b>, and excludes any Interfaces that have <b>VLAN10</b> for the (interface name) <code>ifName</code> value:
<code>(ifDesc like VLAN AND NOT (ifName=VLAN10))</code>
<b>Note</b>: View the expression displayed under Filter String to see the logic of the expression as it is created .</td></tr>
<tr><td>EXISTS</td><td>Used for filters that include Capabilities or Custom Attribute names and values in the Filer String.
Indicates that you want NNMi to consider interfaces that do not have any Capabilities or Custom Attributes when evaluating the Filter String.
<b>Note</b>: If you include Capabilities or Custom Attribute names and values in the Filer String, but do not use <b>EXISTS</b> or <b>NOT EXISTS</b>, NNMi excludes from its search interfaces that do not include Capabilities or Custom Attributes.
For example, when evaluating the following Filter String, NNMi includes interfaces with (interface description) <code>ifDesc</code> containing <b>VLAN</b>, as well as any Interfaces Custom Attribute Role value is <b>LAN Connection to Oracle Server</b>:
<code>(ifDesc like VLAN OR EXISTS((customAttrName=Role AND customAttrValue=LAN Connection to Oracle Server)))</code>
<b>Note</b>: View the expression displayed under Filter String to see the</td></tr>
</table>

</td></tr>
</table>

Node Settings Dampen Attributes , continued

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td></td><td>

Payload Filter Editor Buttons, continued

<table>
<tr><th>Button</th><th>Description</th></tr>
<tr><td></td><td>logic of the expression as it is created.</td></tr>
<tr><td>NOT EXISTS</td><td>Used for filters that include Capabilities or Custom Attribute names and values in the Filer String. Indicates that you want NNMi to consider interfaces that do not have any Capabilities or Custom Attributes when evaluating the Filter String, but exclude the interfaces that match the expression that follows the <b>NOT EXISTS</b>.
<b>Note</b>: If you include Capabilities or Custom Attribute names and values in the Filer String, but do not use <b>EXISTS</b> or <b>NOT EXISTS</b>, NNMi excludes from its search interfaces that do not include Capabilities or Custom Attributes.
For example, when evaluating the following expression, NNMi includes interfaces with (interface description) <code>ifDesc</code> containing <b>VLAN</b>, and excludes any Interfaces that have the Custom Attribute Role and that Role value is <b>LAN Connection to Oracle Server</b>:
<code>(ifDesc like VLAN OR NOT EXISTS ((customAttrName=Role AND customAttrValue=LAN Connection to Oracle Server)))</code>
<b>Note</b>: View the expression displayed under Filter String to see the logic of the expression as it is created.</td></tr>
<tr><td>Delete</td><td>Deletes the selected expression.
<b>Note</b>: If the Boolean Operator is selected, the Payload Filter Editor deletes all expressions associated with the Boolean Operator.</td></tr>
</table>

</td></tr>
</table>

Configure Incident Actions for a Node Group (SNMP Trap Incident)

For information about each Node Settings tab:

Note: Node Settings override any other Actions settings for this incident, except those configured on the Interface Settings tab.

NNMi enables you to configure incident actions based on a Source Node's participation in a Node Group.

You can also configure incident actions based on either of the following:

- The Source Object's participation in an Interface Group. See "Configure Incident Actions for an Interface Group (SNMP Trap Incident)" on page 832 for more information.
- Incident configuration default settings without specifying a Node or Interface Group. See "Configure Actions for an SNMP Trap Incident" on page 913 for more information.

You can configure actions to automatically run at any point in the incident lifecycle. For example, you might want to configure an action to occur when an incident of the type you are configuring is generated (**Registered**). When an incident is generated, you might want to automatically open a trouble ticket or send email or page your network operator. After the incident is **Closed**, you might want to automatically close the trouble ticket.

> **Note:** Your actions will not be executed until you enable the Actions configuration by either clicking Enable ☑ on the Actions tab or using the **Actions** → **Enable Configuration** option.

You can configure actions for incidents generated from SNMP traps and NNMi management events. Any time an incident configuration changes, the action directory is rescanned and any executable or script files (for example, Jython) are reloaded to the NNMi database. See "Lifecycle Transition Action Form (SNMP Trap Incidents)" on page 914 for more information about the actions directory.

Tip: Copy any required executable or script files to the NNMi actions directory before you configure an incident action. New or updated actions are loaded into NNMi only when an incident configuration is updated or created. See "Lifecycle Transition Action Form (SNMP Trap Incidents)" on page 914 for the location of the NNMi action directory.

When the action runs, output is logged to the `incidentActions.*.*.log` file. See "Verify that NNMi Services are Running" on page 85 for more information about log files and where they are located.

To configure an automatic action for an incident:

1. Navigate to the **SNMP Trap Configuration**form.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Expand the **Incidents** folder.
   c. Select **SNMP Trap Configurations**.
   d. Do one of the following:
      i. To create a new incident configuration, click the ✱ New icon.
      ii. To edit an existing incident configuration, select a row, click the 📂 Open icon, and continue.
2. Select the **Node Settings** tab.
3. Do one of the following:
   a. To create a new configuration, click the ✱ New icon.
   b. To edit an existing configuration, select a row, click the 📂 Open icon, and continue.
   c. Make sure you configure the basic Node Setting behavior. See "Configure Node Settings for an SNMP Trap Incident" on page 840 for more information.
4. Select the **Actions** tab.
5. From the **Lifecycle Actions** table toolbar, do one of the following:
   - To create an Action configuration, click the ✱ New icon, and continue.
   - To edit an Action configuration, select a row, click the 📂 Open icon, and continue.

- To delete an Action configuration, select a row and click the Delete icon.

3. In the "Lifecycle Transition Action Form (SNMP Trap Incidents)" on page 914, provide the required information.
4. Click **Save and Close** to save your changes and return to the **SNMP Trap Configuration** form.

   The next time the lifecycle changes, NNMi launches the action associated with the lifecycle for the incident of that type .

Configure a Payload Filter for an Incident Action (Node Settings) (SNMP Trap Incidents)

The Payload Filter Editor enables you to create expressions that further refine the filters used to select the incidents that cause the configured action to run. Make sure to design any complex Payload Filters offline as a Boolean expression first. This method can help to minimize errors when entering your expressions using the Payload Filter editor.

To create a Payload Filter expression:

1. Navigate to the **SNMP Trap Configuration** form:
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Expand the **Incidents** folder.
   c. Select **SNMP Trap Configurations** .
   d. Do one of the following:
      i. To create an incident configuration, click the New icon, and continue.
      ii. To edit an incident configuration, select a row, click the Open icon, and continue.
      iii. To delete an incident configuration, select a row and click the Delete icon.
2. Select the **Node Settings** tab.
3. Do one of the following:
   a. To create a new configuration, click the New icon.
   b. To edit an existing configuration, select a row, click the Open icon, and continue.
   c. To delete an existing configuration, select a row and click the Delete icon.
4. Make sure you configure the basic Node Setting behavior. See "Configure Node Settings for an SNMP Trap Incident" on page 840 for more information.
5. Select the **Actions** tab.
6. Do one of the following:
   a. To create an Action configuration, click the New icon, and continue.
   b. To edit an Action configuration, select a row, click the Open icon, and continue.

   c. To delete an Action configuration, select a row and click the ✖ Delete icon.

7. Make sure the Action Configuration settings are configured. See "Configure Incident Actions for a Node Group (SNMP Trap Incident)" on page 869 for more information.

8. Select the **Payload Filter** tab.

9. Define your Payload Filter (see table). Also see "Guidelines for Creating a Payload Filter".

   a. Plan out the logic needed for your Filter String.

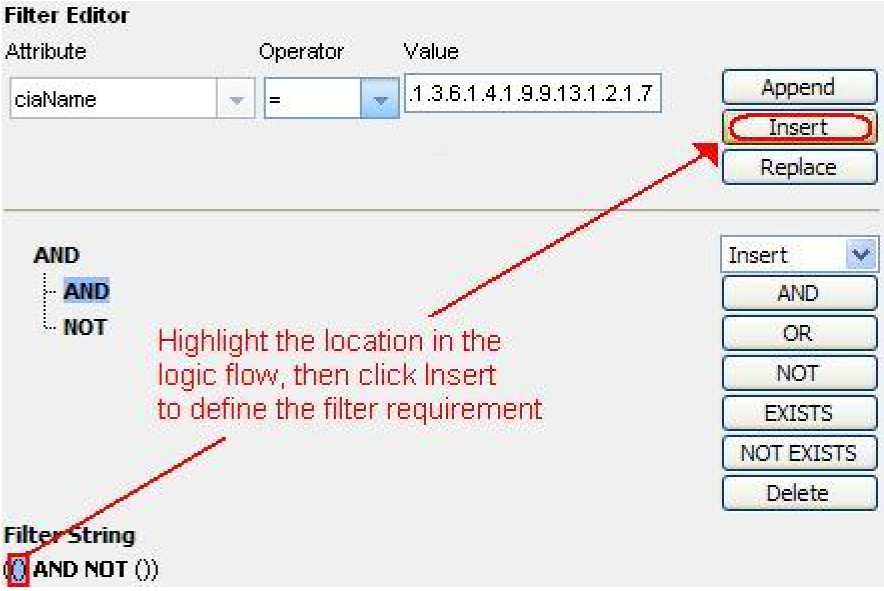
   b. Use the buttons on the bottom half of the Additional Filters Editor to establish the logic structure.

   For example, to establish the following structure, select **Insert**, then click **AND**, then **NOT**, and then **AND** a second time:

   ```
   (( ) AND NOT ( ))
   ```

   c. Now place your cursor in a location within the displayed Filter String, and use the top half of the filter editor to define the parameters of the highlighted filter requirement.

   For example, select a set of parentheses and use the Insert button to specify the filter requirement within those parentheses:



Filter Editor

Attribute: ciaName

Operator: =

Value: .1.3.6.1.4.1.9.9.13.1.2.1.7

Append
Insert
Replace
AND
AND
NOT
Insert
AND
OR
NOT
EXISTS
NOT EXISTS
Delete
Highlight the location in the logic flow, then click Insert to define the filter requirement

Filter String

(() AND NOT ())

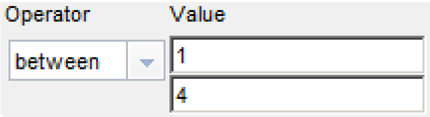
10. Click **Save and Close**.

11. Click **Save and Close** to save your changes and return to the previous form.

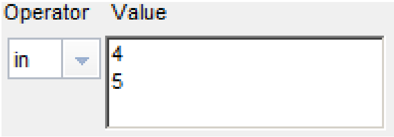
Payload Filter Editor Components

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: • ciaName • ciaValue

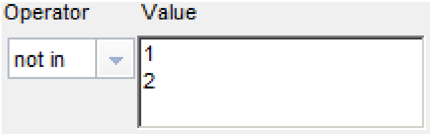
Payload Filter Editor Components, continued

Attribute	Description
Operator	Valid operators are described below. - **=** Finds all values equal to the value specified. Click here for an example. Example: `ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7` matches any incident that contains a varbind with the name value **.1.3.6.1.4.1.9.9.13.1.2.1.7**. - **!=** Finds all values not equal to the value specified. Click here for an example. Example: `ciaName != .1.3.6.1.4.1.9.9.13.1.2.1.7` matches any incident that contains a varbind with the name value other than **1.3.6.1.4.1.9.9.13.1.2.1.7**. - **<** Finds all values less than the value specified. Click here for an example. Example: `ciaValue < 6` matches any incident with a varbind value less than **6**. - **<=** Finds all values less than or equal to the value specified. Click here for an example. Example: `ciaValue <= 6` matches any incident with a varbind value less than or equal to **6**. - **>** Finds all values greater than the value specified. Click here for an example. Example: `ciaValue > 4` matches any incident with a varbind value greater than **4**. - **>=** Finds all values greater than or equal to the value specified. Click here for an example. Example: `ciaValue >= 4` matches any incident that contains a varbind with values greater than or equal to **4**. - **between** Finds all traps or events that include a varbind with a value equal to and between the two values specified. Click here for an example. Example: `ciaValue between`  Operator: between Value: 1 4 matches any incident that contains a varbind value equal to or greater than **1** and equal to or less than **4**. **Note**: As shown in the example, each value must be entered on a separate line. - **in** Finds any match to at least one value in a list of values. Click here for an example. Example: `ciaValue in`

Payload Filter Editor Components, continued

Attribute	Description
	 Operator: in Value: 4 5 matches any incident with a varbind value of either **4** or **5**. **Note**: As shown in the example, each value must be entered on a separate line. NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example (**4, 5**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line. - **is not null** Finds all non-blank values. Click here for an example. Example: `ciaValue is not null` matches any incident with a varbind that contains a value. - **is null** Finds all blank values. Click here for an example. Example: `ciaValue is null` matches any incident with a varbind that does not have a value. - **like** Finds matches using wildcard characters. Click here for more information about using wildcard characters. The period asterisk (.*) characters mean *any number of characters of any type at this location*. The period (.) character means *any single character of any type at this location*. **Note**: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below. Examples: `ciaName like  \Q.1.3.6.1.4.1.9.9\E.*` finds all traps or events that contain varbind names that begin with **.1.3.6.1.4.1.9.9** and (optionally) end with any number of characters. `ciaValue like .*Chicago.*` finds all traps or events that contain a varbind value that includes the string **Chicago**. - **not between** Finds all values except those between the two values specified. Click here for an example. Example: `ciaValue not between 5 8` matches an incident that contains a varbind with the values less than **5** or greater than **8**. - **not in** Finds all values except those included in the list of values. Click here for an example. Example: `ciaValue not in`

Payload Filter Editor Components, continued

Attribute	Description
	 Operator Value not in 1 2 matches any incident that contains a varbind with values other than **1** and **2**. **Note**: As shown in the example, each value must be entered on a separate line. NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example, (**1, 2**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line. • **not like** Finds all that do not have the values specified (using wildcard strings). Click here for an example. The period asterisk (.*) characters mean *any number of characters of any type at this location*. The period (.) character means *any single character of any type at this location*. **Note**: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below. Example: `ciaName not like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that does not begin with **.1.3.6.1.4.1.9.9**. `ciaValue not like .*Chicago.*` finds all traps or events that do not contain a varbind value that includes the string **Chicago**.
Value	The value for which you want NNMi to search. Note the following: • The values you enter are case sensitive. • NNMi displays a variable number of value fields depending on the Operator selected. For example, the `between` Operator causes two value fields to be displayed. • The `between, in` and `not in` operators require that each value be entered on a separate line.

Additional Filters Editor Buttons

Button	Description
Append	Appends the current expression (Attribute, Operator,and Value) to the selected expression already included in the Filter String.
Insert	Inserts the current expression (Attribute, Operator,and Value) in front of the cursor location within the Filter String.

Additional Filters Editor Buttons, continued

Button	Description
Replace	Replaces the selected expression with the expression displayed in the Attribute, Operator, and Value fields.
AND	Appends, inserts, or replaces the AND Boolean Operator in the selected cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
OR	Appends, Inserts, or replaces the OR Boolean Operator in the current cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
NOT	Can be used in any part of the Filter String to specify that NNMi should exclude interfaces with values that pass the expression that immediately follows the `NOT`. For example, when evaluating the following expression, NNMi includes interfaces with (interface description) `ifDesc` containing **VLAN**, and excludes any Interfaces that have **VLAN10** for the (interface name) `ifName` value: `(ifDesc like VLAN AND NOT (ifName=VLAN10))` **Note**: View the expression displayed under Filter String to see the logic of the expression as it is created .
EXISTS	Used for filters that include Capabilities or Custom Attribute names and values in the Filer String. Indicates that you want NNMi to consider interfaces that do not have any Capabilities or Custom Attributes when evaluating the Filter String. **Note**: If you include Capabilities or Custom Attribute names and values in the Filer String, but do not use **EXISTS** or **NOT EXISTS**, NNMi excludes from its search interfaces that do not include Capabilities or Custom Attributes. For example, when evaluating the following Filter String, NNMi includes interfaces with (interface description) `ifDesc` containing **VLAN**, as well as any Interfaces Custom Attribute Role value is **LAN Connection to Oracle Server**: `(ifDesc like VLAN OR EXISTS((customAttrName=Role AND customAttrValue=LAN Connection to Oracle Server)))` **Note**: View the expression displayed under Filter String to see the logic of the expression as it is created.
NOT EXISTS	Used for filters that include Capabilities or Custom Attribute names and values in the Filer String. Indicates that you want NNMi to consider interfaces that do not have any Capabilities or Custom Attributes when evaluating the Filter String, but exclude the interfaces that match the expression that follows the **NOT EXISTS**.

Additional Filters Editor Buttons, continued

Button	Description
	Note: If you include Capabilities or Custom Attribute names and values in the Filer String, but do not use **EXISTS** or **NOT EXISTS**, NNMi excludes from its search interfaces that do not include Capabilities or Custom Attributes. For example, when evaluating the following expression, NNMi includes interfaces with (interface description) `ifDesc` containing **VLAN**, and excludes any Interfaces that have the Custom Attribute Role and that Role value is **LAN Connection to Oracle Server**: `(ifDesc like VLAN OR NOT EXISTS((customAttrName=Role AND customAttrValue=LAN Connection to Oracle Server)))` **Note**: View the expression displayed under Filter String to see the logic of the expression as it is created.
Delete	Deletes the selected expression. **Note**: If the Boolean Operator is selected, the Additional Filters Editor deletes all expressions associated with the Boolean Operator.

Configure Diagnostics Selections for a Node Group (SNMP Trap Incident) (NNM iSPI NET)

For information about each Node Settings tab: .

Note: Node Settings override any other Diagnostics Selections settings for this incident, except those configured on the Interface Settings tab.

(*HP Network Node Manager iSPI Network Engineering Toolset Software*) The Diagnostic Selections form enables you to configure NNMi to automatically gather NNM iSPI NET diagnostic information for the Incident you are configuring. When using this form, you specify the diagnostics you want to run on each applicable node in the specified Node Group.

To configure Diagnostics to run on a Source Node for an incident:

1. Navigate to the **Diagnostics Selection** form.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Expand the **Incidents** folder.
   c. Select **SNMP Trap Configurations** .
   d. Do one of the following:
      - To create an Incident configuration, click the ✱ New icon.
      - To edit an Incident configuration, select a row, click the 📂 Open icon, and continue.
   e. Navigate to **Node Settings** tab, and do one of the following:

- To create a Node Settings configuration, click the New icon.
- To edit a Node Settings configuration, select a row, click the Open icon, and continue.
- To delete a Node Settings configuration, select the Node setting, and click the Delete icon.

f. Navigate to the **Diagnostic Selection** tab, and do one of the following:

- To create a Diagnostic Selection setting, click the New icon, and continue.
- To edit a Diagnostic Selection setting, select a row, click the Open icon, and continue.
- To delete a Diagnostic Selection setting, select a row and click the Delete icon.

2. Provide the required information (see table).
3. Click **Save and Close** to save your changes and return to the **Node Settings** form.

After you configure the Diagnostic for the incident and Node Group, the Diagnostic must match the following criteria before the Diagnostic runs:

- The Source Node must be in the specified Node Group.
- The Diagnostic must be valid for the Source Node. (For example, only Nortel switch Diagnostics are run on Nortel switches.)
- The incident’s current lifecycle state must match a lifecycle state for which it was configured. (For example, if you configure the Incident to run a specified Diagnostic when the incident is Closed, then if the current Incident's Lifecycle State is Closed, NNMi runs that Diagnostic.)

Note: If a Source Node is in more than one Node Group, the Diagnostic is only run on the node the first time NNMi finds a match for that Source Node based on the configuration Ordering field.

If these criteria are met, NNM iSPI NET runs the Diagnostics and generates Diagnostic reports to help you solve the problem on the Source Node.

After you configure Diagnostics for an incident, you can also run Diagnostics and access the Diagnostics reports on demand, using **Actions** → **Run Diagnostics (iSPI NET only)** in the Incident form. The same criteria apply (see the criteria above). See Incident Form:Diagnostics Tab for more information.

> **Tip:** You can right-click any object in a table or map view to access the **Actions** menu.

You can also run and access Diagnostics reports from a Node form. See Node Form: Diagnostics Tab for more information.

Diagnostic Settings Attributes

Attribute	Description
Flow Definition	Select the Diagnostic (Flow Definition) you want to use for the specified Node Group. Click the Lookup icon and choose one of the following options:

Diagnostic Settings Attributes, continued

Attribute	Description
	• Show Analysis to display Analysis Pane information for Diagnostic (Flow Definition). (See Use the Analysis Pane for more information about the Analysis Pane.) • Quick Find to view the list of possible diagnostic Flow Definitions. NNMi provides diagnostics for the following types of devices: • Cisco switch • Cisco router • Cisco switch/router • Nortel switch See "Diagnostics (Flows) Provided by NNM iSPI NET" on page 758 for more information about the diagnostics provided and the devices to which they apply.
Lifecycle State	Incident Lifecycle State of the target Incident. If the incident's Lifecycle State matches the value specified here, the Diagnostic runs. The Diagnostic automatically runs on each applicable Source Node in the specified Node Group if the incident has the Lifecycle State currently configured in this attribute of the Diagnostic (Flow Definition - set of automated commands).
Enable	Use this attribute to temporarily disable an incident's Diagnostics settings: **Enable** □ = Temporarily disable the selected configuration. **Enable** ☑ = Enable the selected configuration.

Configure Suppression Settings for an SNMP Trap Incident

For information about each SNMP Trap tab:

NNMi enables you to suppress incidents based on Interface Group, Node Group, or default Suppression settings. NNMi applies your Suppression settings in the following order. Only the first match applies.

1. Interface Group (SNMP Trap Configuration Form: Interface Settings tab)
2. Node Group (SNMP Trap Configuration Form: Node Settings tab)
3. Suppression configuration settings without specifying an Interface Group or Node Group (SNMP Trap Configuration Form: Suppresion tab)

A Payload Filter enables you to use the data that is included with any of the following items before they are stored as incidents in NNMi:

- Traps generated from an SNMP agent
- Syslog Messages.
- Management incidents that are generated by NNMi.

Examples of the type of data that can be used as a Payload Filter include SNMP trap varbind names and values as well as CIA (Custom Incident Attribute) names and values. For example, you might want NNMi to suppress a particular status change notification trap for a specified Node Group or Interface Group. To do so, you could include the name of the trap varbind that stores this information as well as the particular status change value string the traps that you want to suppress should contain.

See "Configure Incident Suppression Settings for an Interface Group (SNMP Trap Incident)" on page 804 for information about how to suppress an incident for an Interface Group with or without a Payload Filter.

See "Configure Incident Suppression Settings for a Node Group (SNMP Trap Incident)" on page 841 for more information about how to suppress an incident for a Node Group with or without a Payload Filter.

To configure suppression for an incident using a Payload Filter without an Interface Group or Node Group Filter:

1. Navigate to the **SNMP Trap Configuration** form:
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Expand **Incidents**.
   c. Select **SNMP Trap Configurations** .
   d. Do one of the following:
      i. To create an incident configuration, click the ✳ New icon, and continue.
      ii. To edit an incident configuration, click the 📂 Open icon in the row representing the configuration you want to edit, and continue.
      iii. To delete an incident configuration, click the ✖ Delete icon.
2. Select the **Suppression** tab.
3. Provide the required information (see table)
4. Click 💾 **Save and Close** to save your changes and return to the previous form.

Suppression Attributes

Name	Description
Enable	Use this attribute to temporarily disable an incident's suppression settings: **Enable** □ = Temporarily disable the selected configuration. **Enable** ☑ = Enable the selected configuration.
Payload	The Payload Filter Editor enables you to create expressions that further refine the filters

Suppression Attributes , continued

Name	Description

Filter	used to select the incidents to be suppressed, enriched, or dampened. Make sure to design any complex Payload Filters offline as a Boolean expression first. This method can help to minimize errors when entering your expressions using the Payload Filter editor.
	When creating a Payload Filter, note the following:
	• Payload Filter expressions for the `like` and `not like` operators use the syntax defined for java regular expressions (java.util.regex Pattern class).
	• You must use a `ciaName` that already exists in the trap or event you are configuring.
	• Each set of expressions associated with a Boolean Operator is treated as if it were enclosed in parentheses and evaluated together.
	• View the expression displayed under **Filter String** to see the logic of the expression as it is created.
	• The `AND` and `OR` Boolean Operators must contain at least two expressions as shown in the example below.
	The following example filters incidents on voltage state:
	`AND`
	`    ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7`
	`    ciaValue = 5`
	NNMi evaluates the expression above as follows:
	`(ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7 AND ciaValue = 5)`
	NNMi finds all incidents with a varbind `.1.3.6.1.4.1.9.9.13.1.2.1.7` value of **5**.
	• The placement of your cursor and the subsequent text that is selected is important when performing operations using the Payload Filter Editor. For example, you append to, replace, or change the indentation of the expression that is selected.
	• The placement of your cursor and the subsequent text that is selected is especially important when adding your Boolean operators.
	• You can include more than one varbind in the same Payload Filter expression as shown in the following example:
	`((ciaName like \Q.1.3.6.1.4.1.9.9\E.* AND ciaValue = 25) AND (ciaName like \Q.1.3.6.1.2.1.2.2.1.1.3\E.* AND ciaValue = 3))`
	In this example, a given trap must meet each of the following criteria:
	▪ Contain a varbind whose Object Identifier (OID) matches the regular expression `\Q.1.3.6.1.4.1.9.9\E.*` and has a value of `25`.
	▪ Contain a varbind whose OID matches the regular expression `\Q.1.3.6.1.2.1.2.2.1.1.3\E.*` and has a value of `3`.

Suppression Attributes , continued

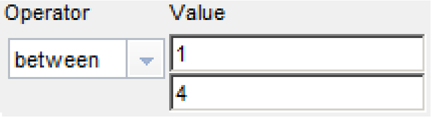
<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td></td><td>
<b>Payload Filter Editor Components</b>
<table>
<tr><th>Attribute</th><th>Description</th></tr>
<tr><td>Attribute</td><td>The attribute name on which NNMi searches. Filterable attributes include the following:
<ul>
<li>ciaName</li>
<li>ciaValue</li>
</ul></td></tr>
<tr><td>Operator</td><td>Valid operators are described below.
<ul>
<li><b>=</b> Finds all values equal to the value specified. Click here for an example.
Example: <code>ciaName=.1.3.6.1.4.1.9.9.13.1.2.1.7</code> matches any incident that contains a varbind with the name value of <b>.1.3.6.1.4.1.9.9.13.1.2.1.7</b>.</li>
<li><b>!=</b> Finds all values not equal to the value specified. Click here for an example.
Example: <code>ciaName! = .1.3.6.1.4.1.9.9.13.1.2.1.7</code> matches any incident that contains a varbind with a name value other than <b>1.3.6.1.4.1.9.9.13.1.2.1.7</b>.</li>
<li><b>&lt;</b> Finds all values less than the value specified. Click here for an example.
Example: <code>ciaValue &lt; 6</code> matches any incident that contains a varbind with a value less than <b>6</b>.</li>
<li><b>&lt;=</b> Finds all values less than or equal to the value specified. Click here for an example.
Example: <code>ciaValue &lt;= 6</code> matches any incident that contains a varbind with a value less than or equal to <b>6</b>.</li>
<li><b>&gt;</b> Finds all values greater than the value specified. Click here for an example.
Example: <code>ciaValue &gt; 4</code> matches any incident that contains a varbind with a value greater than <b>4</b>.</li>
<li><b>&gt;=</b> Finds all values greater than or equal to the value specified. Click here for an example.
Example: <code>ciaValue &gt;= 4</code> matches any incident that contains a varbind with values greater than or equal to <b>4</b>.</li>
<li><b>between</b> Finds all values equal to and between the two values specified. Click here for an example.
Example: <code>ciaValue between</code></li>
</ul></td></tr>
</table>
</td></tr>
</table>

Suppression Attributes , continued

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td></td><td>

Payload Filter Editor Components, continued

<table>
<tr><th>Attribute</th><th>Description</th></tr>
<tr><td></td><td>


Operator: between
Value: 1
4

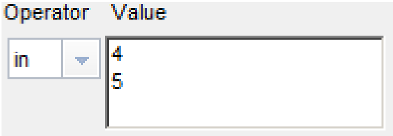
matches any incident that contains a varbind value equal to or greater than **1** and equal to or less than **4**.

Note: As shown in the example, each value must be entered on a separate line.

- **in** Finds any match to at least one value in a list of values. Click here for an example.

Example:

`ciaValue in`


Operator: in
Value: 4
5

matches any incident with a varbind value of either **4** or **5**.

Note: As shown in the example, each value must be entered on a separate line.

NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example (**4, 5**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line.

- **is not null** Finds all non-blank values. Click here for an example.

Example: `ciaValue is not null` matches any incident with a varbind that contains a value.

- **is null** Finds all blank values. Click here for an example.

Example: `ciaValue is null` matches any incident with a varbind that does not contain a value.

- **like** Finds matches using the syntax defined for java regular expressions. See the Pattern (Java Platform SE6) API documentation at : `http://download.oracle.com/javase/6/docs/api/java/util/regex/Pattern.html` for more information. Click here for more information.

</td></tr>
</table>

</td></tr>
</table>

Suppression Attributes , continued

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td></td><td>

Payload Filter Editor Components, continued

<table>
<tr><th>Attribute</th><th>Description</th></tr>
<tr><td></td><td>

The period asterisk (.*) characters mean *any number of characters of any type at this location*.

The period (.) character means *any single character of any type at this location*.

Note: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below.

Example:

`ciaName like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that begins with 1.3.6.1.4.1.9.9 and (optionally) ends with any number of characters.

`ciaValue like .*Chicago.*` finds all traps or events that contain a varbind value that includes the string **Chicago**.

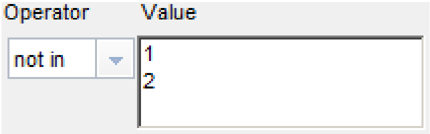
- **not between** Finds all values except those between the two values specified. Click here for an example.

  Example: `ciaValue not between 5 8` matches an incident that contains a varbind with the values less than **5** or greater than **8** .

- **not in** Finds all values except those included in the list of values. Click here for an example.

  Example:

  `ciaValue not in`

  Operator: not in
  Value:
  1
  2

  matches any incident that contains a varbind with values other than **1** and **2**.

  Note: As shown in the example, each value must be entered on a separate line.

  NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example, (**1, 2**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line.

</td></tr>
</table>

</td></tr>
</table>

Suppression Attributes , continued

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td></td><td>

<b>Payload Filter Editor Components, continued</b>

<table>
<tr><th>Attribute</th><th>Description</th></tr>
<tr><td></td><td>
<ul><li><b>not like</b> Finds all that do not have the values specified using the syntax defined for Java regular expressions. See the Pattern (Java Platform SE6) API documentation at: <code>http://download.oracle.com/javase/6/docs/api/java/util/regex/Pattern.html</code> for more information. Click here for an example.

<p>The period asterisk (.*) characters mean <i>any number of characters of any type at this location</i>.</p>
<p>The period (.) character means <i>any single character of any type at this location</i>.</p>
<p><b>Note</b>: To include literal string values in the Value attribute, enclose the string value in \Q&lt;<i>literal_value</i>&gt;\E as shown in the Examples listed below.</p>
<p>Example:</p>
<p><code>ciaName not like \Q.1.3.6.1.4.1.9.9\E.*</code> matches any incident that contains a varbind name value that does not begin with <b>.1.3.6.1.4.1.9.9</b> and (optionally) ends with any number of characters.</p>
<p><code>ciaValue not like .*Chicago.*</code> finds all traps or events that do not contain a varbind value that includes the string <b>Chicago</b>.</p></li></ul>
</td></tr>
<tr><td>Value</td><td>
<p>The value for which you want NNMi to search.</p>
<p>Note the following:</p>
<ul>
<li>The values you enter are case sensitive.</li>
<li>NNMi displays a variable number of value fields depending on the Operator selected. For example, the <code>between</code> Operator causes two value fields to be displayed.</li>
<li>The <code>between, in</code> and <code>not in</code> operators require that each value be entered on a separate line.</li>
</ul>
</td></tr>
</table>

<b>Payload Filter Editor Buttons</b>

<table>
<tr><th>Button</th><th>Description</th></tr>
<tr><td>Append</td><td>Appends the current expression (Attribute, Operator,and Value) to the selected expression already included in the filter string.</td></tr>
</table>

</td></tr>
</table>

Suppression Attributes , continued

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td></td><td>

Payload Filter Editor Buttons, continued

<table>
<tr><th>Button</th><th>Description</th></tr>
<tr><td>Insert</td><td>Inserts the current expression (Attribute, Operator,and Value) in front of the cursor location within the Filter String.</td></tr>
<tr><td>Replace</td><td>Replaces the selected expression with the expression displayed in the Attribute, Operator, and Value fields.</td></tr>
<tr><td>AND</td><td>Inserts the AND Boolean Operator in the selected cursor location.
Note: View the expression displayed under **Filter String** to see the logic of the expression as it is created.</td></tr>
<tr><td>OR</td><td>Inserts the OR Boolean Operator in the current cursor location.
Note: View the expression displayed under **Filter String** to see the logic of the expression as it is created.</td></tr>
<tr><td>NOT</td><td>Can be used in any part of the Filter String to specify that NNMi should exclude interfaces with values that pass the expression that immediately follows the `NOT`.
For example, when evaluating the following expression, NNMi includes interfaces with (interface description) `ifDesc` containing **VLAN**, and excludes any Interfaces that have **VLAN10** for the (interface name) `ifName` value:
`(ifDesc like VLAN AND NOT (ifName=VLAN10))`
Note: View the expression displayed under Filter String to see the logic of the expression as it is created .</td></tr>
<tr><td>EXISTS</td><td>Used for filters that include Capabilities or Custom Attribute names and values in the Filer String.
Indicates that you want NNMi to consider interfaces that do not have any Capabilities or Custom Attributes when evaluating the Filter String.
Note: If you include Capabilities or Custom Attribute names and values in the Filer String, but do not use **EXISTS** or **NOT EXISTS**, NNMi excludes from its search interfaces that do not include Capabilities or Custom Attributes.
For example, when evaluating the following Filter String, NNMi includes interfaces with (interface description) `ifDesc` containing **VLAN**, as well as any Interfaces Custom Attribute Role value is **LAN Connection to Oracle Server**:</td></tr>
</table>

</td></tr>
</table>

Suppression Attributes , continued

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td></td><td>

Payload Filter Editor Buttons, continued

<table>
<tr><th>Button</th><th>Description</th></tr>
<tr><td></td><td><code>(ifDesc like VLAN OR EXISTS((customAttrName=Role AND customAttrValue=LAN Connection to Oracle Server)))</code>

<b>Note</b>: View the expression displayed under Filter String to see the logic of the expression as it is created.</td></tr>
<tr><td>NOT EXISTS</td><td>Used for filters that include Capabilities or Custom Attribute names and values in the Filer String. Indicates that you want NNMi to consider interfaces that do not have any Capabilities or Custom Attributes when evaluating the Filter String, but exclude the interfaces that match the expression that follows the <b>NOT EXISTS</b>.

<b>Note</b>: If you include Capabilities or Custom Attribute names and values in the Filer String, but do not use <b>EXISTS</b> or <b>NOT EXISTS</b>, NNMi excludes from its search interfaces that do not include Capabilities or Custom Attributes.

For example, when evaluating the following expression, NNMi includes interfaces with (interface description) <code>ifDesc</code> containing <b>VLAN</b>, and excludes any Interfaces that have the Custom Attribute Role and that Role value is <b>LAN Connection to Oracle Server</b>:

<code>(ifDesc like VLAN OR NOT EXISTS ((customAttrName=Role AND customAttrValue=LAN Connection to Oracle Server)))</code>

<b>Note</b>: View the expression displayed under Filter String to see the logic of the expression as it is created.</td></tr>
<tr><td>Delete</td><td>Deletes the selected expression.

<b>Note</b>: If the Boolean Operator is selected, the Payload Filter Editor deletes all expressions associated with the Boolean Operator.</td></tr>
</table>

</td></tr>
</table>

Configure Enrichment Settings for an SNMP Trap Incident

For information about each SNMP Traps tab:

For information about each Enrichment tab:

NNMi enables you to fine tune and enhance incidents based on Interface Group, Node Group, or default Enrichment settings. NNMi applies your Enrichment settings in the following order. Only the first match applies:

1. Interface Group (SNMP Trap Configuration Form: Interface Settings tab)
2. Node Group (SNMP Trap Configuration Form: Node Settings tab)
3. Enrich configuration settings without specifying an Interface Group or Node Group (SNMP Trap Configuration Form: Enrichment tab)

The types of items you can fine tune and enhance for a selected incident configuration, include:

- Category
- Family
- Severity
- Priority
- Correlation Nature
- Message
- Assigned To

Note: Any configuration you specify for Severity, Priority, or Message overrides those values provided in the SNMP Trap Configuration Form: Basics information.

You can also add a Custom Incident Attribute that is provided by NNMi to the incoming incident.

Note: You cannot create Custom Incident Attributes.

When configuring Interface Settings, Node Settings, or other Suppress Configuration, Enrich Configuration, or Dampening configuration settings for an incident, you can specify a Payload Filter. Payload Filters enables you to use the data that is included with any of the following items before they are stored as incidents in NNMi:

- Traps generated from an SNMP agent
- Management incidents that are generated by NNMi
- Events generated by NNM 6.x or 7.x management stations

Examples of the type of data that can be used as a Payload Filter include SNMP trap varbind names and values as well as Management Event CIA (Custom Incident Attribute) names and values. For example, you might want NNMi to enrich an incident based on a particular status change notification trap and participation within a specified Node Group or Interface Group. To do so, you would first specify participation in the Node Group or Interface Group for the trap you want to enrich. You would also specify a Payload Filter that includes the name of the trap varbind that stores the status information as well as the status change value string of interest.

See "Configure Incident Enrichment Settings for an Interface Group (SNMP Trap Incident)" on page 812 for information about how to enrich an incident for an Interface Group with or without a Payload Filter.

See "Configure Incident Enrichment Settings for a Node Group (SNMP Trap Incident)" on page 849 for more information about how to enrich an incident for a Node Group with or without a Payload Filter.

To configure Enrichment settings for an incident using a Payload Filter without an Interface Group or Node Group Filter:

1. Navigate to the **SNMP Trap Configuration** form:
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Expand the **Incidents** folder.
   c. Select **SNMP Trap Configurations** .
   d. Do one of the following:
      i. To create an incident configuration, click the New icon, and continue.
      ii. To edit an incident configuration, select a row, click the Open icon, and continue.
      iii. To delete an incident configuration, select a row and click the Delete icon.
2. Select the **Enrichment** tab.
3. Do one of the following:
   a. To create a new configuration, click the New icon.
   b. To edit an existing configuration, select a row, click the Open icon, and continue.
4. Provide the required information (see table)
5. Click **Save and Close** to save your changes and return to the previous form.

Enrichment Attributes

Name	Description
Category	Use the Category attribute to customize the category for this incident configuration. Possible values include: • Accounting • Application Status • Configuration • Fault • Performance • Security • Status See "Specify Category and Family Attribute Values for Organizing Your Incidents (SNMP Trap Incident)" on page 790 for more information.
Family	Use the Family attribute to customize the Family for this incident configuration. Select from the drop-down list or create a new value. For example, some of the values provided by NNMi include: • Address

Enrichment Attributes , continued

Name	Description
	• Aggregated Port (Interfaces using **Link Aggregation**[1] protocol. See Interface Form: Link Aggregation tab.) • Card • Connection • Correlation • Interface • Node
Severity	The incident Severity represents the seriousness calculated for the incident. Use the Severity attribute to specify the severity that should be assigned to the incident you are configuring. Possible values are described below: **Normal** - Indicates there are no known problems related to the associated object. This Severity is meant to be informational. Generally, no action is needed for these incidents. **Warning** - Indicates there might be a problem related to the associated object. **Minor** - Indicates NNMi has detected problems related to the associated object that require further investigation. **Major** - Indicates NNMi has detected problems related to the associated object to be resolved before they become critical. **Critical** - Indicates NNMi has detected problems related to the associated object that require immediate attention.
Priority	Used to communicate the urgency of resolving the selected incident. You control this value. NNMi sets this value to null by default. The lower the number the higher the priority. Possible values are: 5 **None** 4 **Low** 3 **Medium** 2 **High**

[1] Protocols used on Switches to configure multiple Interfaces (Aggregation Member Interfaces) to function as if they were one (an Aggregator Interface). When two Aggregator Interfaces establish a connection, that connection is an Aggregator Layer 2 Connection. The Aggregator Layer 2 Connection appears on Layer 2 Neighbor View maps as a thick line with an Interface icon at each end (representing the Aggregator Interface). When you double-click the thick line, it converts into multiple thin lines representing the participating Aggregation Member Layer 2 Connections with their Aggregation Member Interfaces at each end of the lines.

Enrichment Attributes , continued

Name	Description
	Top **Note**: The icons are displayed only in table views.
Correlation Nature	Use the Correlation Nature to customize the Correlation Nature for this incident configuration. Possible values include: • Info • None • Root Cause • Secondary Root Cause • Symptom • Stream Correlation • Service Impact • Dedup Stream Correlation • Rate Stream Correlation See Incident Form: General Tab for more information.
Message Format	When configuring an incident, specify how the incident message appears in the incident view. The string you specify in the Message Format attribute is visible in an incident view. **Note**: The incident message limit is 1024 characters. If you exceed this limit, NNMi truncates the value starting from the right. You can use any combination of default and custom attributes: "Valid Parameters for Configuring Incident Messages (SNMP Trap Incident)" on page 795 "Include Custom Incident Attributes in Your Message Format (SNMP Trap Incident)" on page 801
Assigned To	Use to specify the owner of any incident generated for this incident configuration. Click the ▾ Lookup icon and select Quick Find to select a valid user name. **Note**: You can also begin to type a valid user name and use the auto-complete feature to select the user name of interest.
Description	Use the Description attribute to provide additional information that you want to note about the current enhancement configuration. This description applies only to the enhancement configuration and does not appear when NNMi displays any associated incident. Type a maximum of 1024 characters. Alpha-numeric and special characters (~ ! @ # $ % ^ & * ( ) _+ -) are permitted.

Configure Dampening Settings for an SNMP Trap Incident

For information about each SNMP Traps tab:

NNMi enables you to delay (dampen) the following for an incident configuration:

- Appearance within Incident views in the NNMi Console
- Execution of Incident Actions
- Execution of Diagnostics (*HP Network Node Manager iSPI Network Engineering Toolset Software \ NNM iSPI NET*)

You can configure Dampening settings based on Interface Group, Node Group, or default Dampening settings. NNMi applies your Dampening settings in the following order. Only the first match applies.

1. Interface Group (SNMP Trap Configuration Form: Interface Settings tab)
2. Node Group (SNMP Trap Configuration Form: Node Settings tab)
3. Dampening configuration settings without specifying an Interface Group or Node Group (SNMP Trap Configuration Form: Dampening tab)

When using Dampening configuration, note the following:

- For all Incident Configurations except Deduplication and Rate Incidents, if the dampened Incident is Closed before the Dampen Interval has passed, NNMi deletes the Incident. If the Incident is the Root Cause Incident, NNMi also deletes any Child Incidents

  Note: NNMi administrators can view the number of incidents Closed and deleted while dampened. Access the **Help** → **System Information** → **Health** tab, click the View Detailed Health Report button, and search for the word dampened.

- For all Incident Configurations except Deduplication and Rate Incidents, if the dampened Incident is Closed before the Dampen Interval has passed, NNMi deletes the Incident. If the Incident is the Root Cause Incident, NNMi also deletes any Child Incidents.
- NNMi always retains the Parent Deduplication or Rate Incident even If its Child Incidents are Closed within the Dampen Interval and subsequently deleted. See "Correlate Duplicate Incidents (Deduplication Configuration)" on page 659 and "Track Incident Frequency (Rate: Time Period and Count)" on page 659 for more information about Duplicate and Rate Correlation incidents.
- Any Deduplication and Incidents that have Child Incidents inherit the Dampening settings from their Correlated Children.
- If an incident is a Root Cause Incident and a Child Incident's Dampen Interval is less than the Parent Incident's Dampen Interval, NNMi holds any Child Incidents until the Dampen Interval for the Parent Incident has passed or until the Parent Incident is Closed and subsequently deleted.
- To make sure NNMi handles both Incidents in a Pairwise Configuration the same, configure the same Dampen Interval for each Incident in a Pairwise Incident Configuration.
- After the Dampen Interval has passed, NNMi changes the Lifecycle State to REGISTERED.

See About the Incident Lifecycle for more information about Lifecycle State.

- You can use a Payload Filter to fine tune the incidents you want to dampen.

When configuring Interface Settings, Node Settings, or other Suppress Configuration, Enrich Configuration, or Dampening configuration settings for an incident, you can specify a Payload Filter. Payload Filters enables you to use the data that is included with any of the following items before they are stored as incidents in NNMi:

- Traps generated from an SNMP agent
- Management incidents that are generated by NNMi
- Events generated by NNM 6.x or 7.x management stations

Examples of the type of data that can be used as a Payload Filter include SNMP trap varbind names and values as well as Management Event CIA (Custom Incident Attribute) names and values. For example, you might want NNMi to dampen an incident based on a particular status change notification trap and participation within a specified Node Group or Interface Group. To do so, you would first specify participation in the Node Group or Interface Group for the trap you want to dampen. You would also specify a Payload Filter that includes the name of the trap varbind that stores the status information as well as the status change value string of interest.

See "Configure Incident Dampening Settings for an Interface Group (SNMP Trap Incident)" on page 824 for information about how to configure Dampening settings for an Interface Group with or without a Payload Filter.

See "Configure Incident Dampening Settings for a Node Group (SNMP Trap Incident)" on page 861 for more information about how to configure Dampening for a Node Group with or without a Payload Filter.

To configure Dampening settings for an incident using a Payload Filter without an Interface Group or Node Group Filter:

1. Navigate to the **SNMP Trap Configuration** form:
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Expand the **Incidents** folder.
   c. Select **SNMP Trap Configurations**.
   d. Do one of the following:
      i. To create a configuration, click the ✳ New icon, and continue.
      ii. To edit configuration, select a row, click the 📂 Open icon, and continue.
      iii. To delete a configuration, select a row and click the ✖ Delete icon.
2. Select the **Dampening** tab.
3. Provide the required information (see table)
4. Click 💾 **Save and Close** to save your changes and return to the previous form.

Dampening Attributes

Name	Description
Enable	Use this attribute to temporarily disable an incident's Dampening settings: **Enable** □ = Temporarily disable the selected configuration. **Enable** ☑ = Enable the selected configuration.
Hour	Specifies the number of hours to be used for the Dampen Interval.
Minutes	Specifies the number of minutes to be used for the Dampen Interval.
Seconds	Specifies the number of seconds to be used for the Dampen Interval.

Payload Filter	The Payload Filter Editor enables you to create expressions that further refine the filters used to select the incidents to be suppressed, enriched, or dampened. Make sure to design any complex Payload Filters offline as a Boolean expression first. This method can help to minimize errors when entering your expressions using the Payload Filter editor.
	When creating a Payload Filter, note the following:
	▪ Payload Filter expressions for the `like` and `not like` operators use the syntax defined for java regular expressions (java.util.regex Pattern class).
	▪ You must use a `ciaName` that already exists in the trap or event you are configuring.
	▪ Each set of expressions associated with a Boolean Operator is treated as if it were enclosed in parentheses and evaluated together.
	▪ View the expression displayed under **Filter String** to see the logic of the expression as it is created.
	▪ The `AND` and `OR` Boolean Operators must contain at least two expressions as shown in the example below.
	The following example filters incidents on voltage state:
	`AND`
	`    ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7`
	`    ciaValue = 5`
	NNMi evaluates the expression above as follows:
	`(ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7 AND ciaValue = 5)`
	NNMi finds all incidents with a varbind `.1.3.6.1.4.1.9.9.13.1.2.1.7` value of **5**.
	▪ The placement of your cursor and the subsequent text that is selected is important when performing operations using the Payload Filter Editor. For example, you append to, replace, or change the indentation of the expression that is selected.

Dampening Attributes , continued

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td></td><td>

- The placement of your cursor and the subsequent text that is selected is especially important when adding your Boolean operators.
- You can include more than one varbind in the same Payload Filter expression as shown in the following example:

```
((ciaName like \Q.1.3.6.1.4.1.9.9\E.* AND ciaValue = 25)
AND (ciaName like \Q.1.3.6.1.2.1.2.2.1.1.3\E.* AND
ciaValue = 3))
```

  In this example, a given trap must meet each of the following criteria:

  - Contain a varbind whose Object Identifier (OID) matches the regular expression `\Q.1.3.6.1.4.1.9.9\E.*` and has a value of `25`.
  - Contain a varbind whose OID matches the regular expression `\Q.1.3.6.1.2.1.2.2.1.1.3\E.*` and has a value of `3`.

Payload Filter Editor Components

<table>
<tr><th>Attribute</th><th>Description</th></tr>
<tr><td>Attribute</td><td>

The attribute name on which NNMi searches. Filterable attributes include the following:

- ciaName
- ciaValue

</td></tr>
<tr><td>Operator</td><td>

Valid operators are described below.

- **=** Finds all values equal to the value specified. Click here for an example.

  Example: `ciaName=.1.3.6.1.4.1.9.9.13.1.2.1.7` matches any incident that contains a varbind with the name value of **.1.3.6.1.4.1.9.9.13.1.2.1.7**.

- **!=** Finds all values not equal to the value specified. Click here for an example.

  Example: `ciaName! = .1.3.6.1.4.1.9.9.13.1.2.1.7` matches any incident that contains a varbind with a name value other than **1.3.6.1.4.1.9.9.13.1.2.1.7**.

- **<** Finds all values less than the value specified. Click here for an example.

  Example: `ciaValue < 6` matches any incident that contains a varbind with a value less than **6**.

</td></tr>
</table>

</td></tr>
</table>

Dampening Attributes , continued

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td></td><td>

Payload Filter Editor Components, continued

<table>
<tr><th>Attribute</th><th>Description</th></tr>
<tr><td></td><td>

- **<=** Finds all values less than or equal to the value specified. Click here for an example.

  Example: `ciaValue <= 6` matches any incident that contains a varbind with a value less than or equal to **6**.

- **>** Finds all values greater than the value specified. Click here for an example.

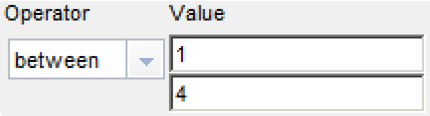
  Example: `ciaValue > 4` matches any incident that contains a varbind with a value greater than **4**.

- **>=** Finds all values greater than or equal to the value specified. Click here for an example.

  Example: `ciaValue >= 4` matches any incident that contains a varbind with values greater than or equal to **4**.

- **between** Finds all values equal to and between the two values specified. Click here for an example.

  Example: `ciaValue between`

  Operator: between
Value: 1
4

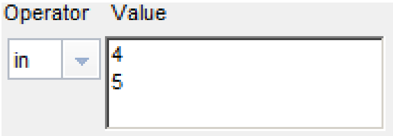
  matches any incident that contains a varbind value equal to or greater than **1** and equal to or less than **4**.

  Note: As shown in the example, each value must be entered on a separate line.

- **in** Finds any match to at least one value in a list of values. Click here for an example.

  Example:

  `ciaValue in`

  Operator: in
Value: 4
5

  matches any incident with a varbind value of either **4** or **5**.

  Note: As shown in the example, each value must be entered on a separate line.

</td></tr>
</table>

</td></tr>
</table>

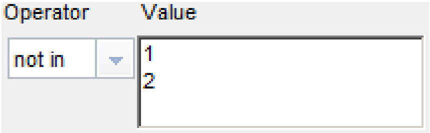
Dampening Attributes , continued

Name	Description
	Payload Filter Editor Components, continued

Attribute	Description
	NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example (**4, 5**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line. ■ **is not null** Finds all non-blank values. Click here for an example. Example: `ciaValue is not null` matches any incident with a varbind that contains a value. ■ **is null** Finds all blank values. Click here for an example. Example: `ciaValue is null` matches any incident with a varbind that does not contain a value. ■ **like** Finds matches using the syntax defined for java regular expressions. See the Pattern (Java Platform SE6) API documentation at : `http://download.oracle.com/javase/6/docs/api/java/util/regex/Pattern.html` for more information. Click here for more information. The period asterisk (.*) characters mean *any number of characters of any type at this location*. The period (.) character means *any single character of any type at this location*. **Note**: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below. Example: `ciaName like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that begins with 1.3.6.1.4.1.9.9 and (optionally) ends with any number of characters. `ciaValue like .*Chicago.*` finds all traps or events that contain a varbind value that includes the string **Chicago**. ■ **not between** Finds all values except those between the two values specified. Click here for an example. Example: `ciaValue not between 5 8` matches an incident that contains a varbind with the values less than **5** or greater than **8** .

Dampening Attributes , continued

Name	Description
	Payload Filter Editor Components, continued

Attribute	Description
	■ **not in** Finds all values except those included in the list of values. Click here for an example. Example: `ciaValue not in`  Operator: not in Value: 1 2 matches any incident that contains a varbind with values other than **1** and **2**. **Note**: As shown in the example, each value must be entered on a separate line. NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example, (**1, 2**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line. ■ **not like** Finds all that do not have the values specified using the syntax defined for Java regular expressions. See the Pattern (Java Platform SE6) API documentation at: `http://download.oracle.com/javase/6/docs/api/java/util/regex/Pattern.html` for more information. Click here for an example. The period asterisk (.*) characters mean *any number of characters of any type at this location*. The period (.) character means *any single character of any type at this location*. **Note**: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below. Example: `ciaName not like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that does not begin with **.1.3.6.1.4.1.9.9** and (optionally) ends with any number of characters.

Dampening Attributes , continued

Name	Description

Payload Filter Editor Components, continued

Attribute	Description
	`ciaValue not like .*Chicago.*` finds all traps or events that do not contain a varbind value that includes the string **Chicago**.
Value	The value for which you want NNMi to search. Note the following: ▪ The values you enter are case sensitive. ▪ NNMi displays a variable number of value fields depending on the Operator selected. For example, the `between` Operator causes two value fields to be displayed. ▪ The `between, in` and `not in` operators require that each value be entered on a separate line.

Payload Filter Editor Buttons

Button	Description
Append	Appends the current expression (Attribute, Operator,and Value) to the selected expression already included in the filter string.
Insert	Inserts the current expression (Attribute, Operator,and Value) in front of the cursor location within the Filter String.
Replace	Replaces the selected expression with the expression displayed in the Attribute, Operator, and Value fields.
AND	Inserts the AND Boolean Operator in the selected cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
OR	Inserts the OR Boolean Operator in the current cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
NOT	Can be used in any part of the Filter String to specify that NNMi should exclude interfaces with values that pass the expression that immediately follows the `NOT`. For example, when evaluating the following expression, NNMi includes interfaces with (interface description) `ifDesc` containing

Dampening Attributes , continued

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td></td><td>
<b>Payload Filter Editor Buttons, continued</b>
<table>
<tr><th>Button</th><th>Description</th></tr>
<tr><td></td><td><b>VLAN</b>, and excludes any Interfaces that have <b>VLAN10</b> for the (interface name) <code>ifName</code> value:
<code>(ifDesc like VLAN AND NOT (ifName=VLAN10))</code>
<b>Note</b>: View the expression displayed under Filter String to see the logic of the expression as it is created .</td></tr>
<tr><td>EXISTS</td><td>Used for filters that include Capabilities or Custom Attribute names and values in the Filer String.
Indicates that you want NNMi to consider interfaces that do not have any Capabilities or Custom Attributes when evaluating the Filter String.
<b>Note</b>: If you include Capabilities or Custom Attribute names and values in the Filer String, but do not use <b>EXISTS</b> or <b>NOT EXISTS</b>, NNMi excludes from its search interfaces that do not include Capabilities or Custom Attributes.
For example, when evaluating the following Filter String, NNMi includes interfaces with (interface description) <code>ifDesc</code> containing <b>VLAN</b>, as well as any Interfaces Custom Attribute Role value is <b>LAN Connection to Oracle Server</b>:
<code>(ifDesc like VLAN OR EXISTS ((customAttrName=Role AND customAttrValue=LAN Connection to Oracle Server)))</code>
<b>Note</b>: View the expression displayed under Filter String to see the logic of the expression as it is created.</td></tr>
<tr><td>NOT EXISTS</td><td>Used for filters that include Capabilities or Custom Attribute names and values in the Filer String. Indicates that you want NNMi to consider interfaces that do not have any Capabilities or Custom Attributes when evaluating the Filter String, but exclude the interfaces that match the expression that follows the <b>NOT EXISTS</b>.
<b>Note</b>: If you include Capabilities or Custom Attribute names and values in the Filer String, but do not use <b>EXISTS</b> or <b>NOT EXISTS</b>, NNMi excludes from its search interfaces that do not include Capabilities or Custom Attributes.
For example, when evaluating the following expression, NNMi</td></tr>
</table>
</td></tr>
</table>

Dampening Attributes , continued

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td></td><td>
<b>Payload Filter Editor Buttons, continued</b>
<table>
<tr><th>Button</th><th>Description</th></tr>
<tr><td></td><td>includes interfaces with (interface description) <code>ifDesc</code> containing <b>VLAN</b>, and excludes any Interfaces that have the Custom Attribute Role and that Role value is <b>LAN Connection to Oracle Server</b>:
<code>(ifDesc like VLAN OR NOT EXISTS ((customAttrName=Role AND customAttrValue=LAN Connection to Oracle Server)))</code>
<b>Note</b>: View the expression displayed under Filter String to see the logic of the expression as it is created.</td></tr>
<tr><td>Delete</td><td>Deletes the selected expression.
<b>Note</b>: If the Boolean Operator is selected, the Payload Filter Editor deletes all expressions associated with the Boolean Operator.</td></tr>
</table>
</td></tr>
</table>

Configure Deduplication for an SNMP Trap Incident

For information about each SNMP Traps tab:

The deduplication configuration determines what values NNMi should match to detect when an SNMP trap, management event, or remote NNM 6.x/7.x event is a duplicate.

Note the following:

- Suppression, Enrichment, and Dampening are not supported for Deduplication incidents.
- NNMi applies only one deduplication configuration per incident. If NNMi generates an incident using a specified deduplication configuration, NNMi continues to correlate duplicate incidents using the original configuration. To use a different deduplication configuration for an incident, first delete the current deduplication incident (created using the original deduplication configuration). NNMi generates the next deduplication incident according to the new deduplication configuration settings.
- By default, NNMi updates the **Duplicate Count** every 30 seconds. This interval cannot be changed.
- NNMi continues to update the duplicate count regardless of an incident's lifecycle state. For example, if an incident's **Lifecycle State** is set to **Closed**, the duplicate count continues to be incremented. See About the Incident Lifecycle for more information. This behavior helps you identify situations in which the incident is not yet fixed. Take note if the Duplicate Count is incremented after a lengthy time period has elapsed, which might indicate there is a new problem with the node, interface, or address.

- Each time you stop and restart ovjboss, any incidents that have not yet been correlated or persisted are lost. This means that after a restart of ovjboss, an incoming incident might not be correlated as expected. For example, after a restart of ovjboss, a duplicate incident might not be correlated under its original parent incident. Instead, a new parent incident might be generated. See "Stop or Start an NNMi Process" on page 82 for more information about starting and stopping the ovjboss process.
- If a Duplicate Correlation Incident is dampened, note the following:
  - Duplicate Correlation Incidents inherit the Dampening settings from its Correlated Children.
  - NNMi always retains the Parent Duplicate Correlation incident, even if its Child Incidents are Closed and subsequently deleted.

    See "Dampening Incident Configurations" on page 679 for more information about Dampening an incident configuration.

To specify or delete a deduplication configuration:

1. Navigate to the **SNMP Trap Configuration** form:
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Click to expand the **Incidents** folder.
   c. Select **SNMP Trap Configurations.**
   d. Do one of the following:
      i. To create a deduplication configuration, click the New icon, and continue.
      ii. To edit a deduplication configuration, select a row, click the Open icon, and continue.
      iii. To delete a deduplication configuration, select a row and click the Delete icon.
2. Select the **Deduplication** tab.
3. Provide the required information (see "Deduplication Attributes" table).
4. Click **Save and Close** to save your changes and return to the previous form.

Deduplication Attributes

Name	Description
Enabled	Use this attribute to temporarily disable an incident's deduplication configuration: **Enable** □ = Temporarily disable the selected configuration. **Enable** ☑ = Enable the selected configuration. **Note:** After a deduplication configuration is enabled, NNMi increments the **Duplicate Count** for an associated incident regardless of the **Lifecycle State** value. For example, if an incident's **Lifecycle State** is set to **Closed**, the duplicate count continues to be incremented. See About the Incident Lifecycle for more information.
Count	Specifies the number of duplicate incidents for the current configuration that

Deduplication Attributes, continued

Name	Description
	NNMi stores at one time. For example, if the Count is 10, after NNMi receives 10 duplicate incidents, NNMi deletes the first (oldest) duplicate incident and keeps the eleventh. (NNMi stores ten maximum.)
Hours	Used with the Minute and Second Intervals to specify the time that must elapse before a new duplicate incident is generated for this incident configuration. For example, if the Hour Interval value is 1, and no Minute or Second Intervals are specified, and the duplicate incident is not generated within one hour, NNMi generates a new duplicate incident the next time it occurs.
Minutes	Used with the Hour and Second interval to specify the time that must elapse before a new duplicate incident is generated for this incident configuration. For example, if the Minute Interval is 30 and no Hour or Second Intervals are specified, and the duplicate incident is not generated within 30 minutes, NNMi generates a new duplicate incident the next time it occurs.
Seconds	Used with the Hour and Minute Intervals to specify the time that must elapse before a new duplicate incident is generated for this incident configuration. For example, if the Second Interval is 120 and no Hour or Minute Intervals are specified, and the duplicate incident is not generated within 120 seconds, NNMi generates a new duplicate incident the next time it occurs.
Parent Incident	varUsed to specify the Incident Configuration that will be the Parent Incident for the incident you are configuring. For example, you might have created a Management Event Incident Configuration that could be used as the **Parent Incident** for SNMP Trap Incidents. When specifying the **Parent Incident**, you have the following options: • When you want to use a configuration that NNMi provides, use the default **Duplicate Correlation** incident configuration . If you select this option, the incident message for the Parent Incident begins as follows: `Duplicate Correlation for` *incident_configuration_name*> For example if you are configuring a **Node Down** incident and select **Duplicate Correlation** as the **Parent Incident**, the Parent Incident message begins with: **Duplicate Correlation for Node Down**. Each **Node Down** incident that is a duplicate then appears correlated under the **Duplicate Correlation for Node Down** incident. • NNMi also enables you to customize the Parent Incident for a given deduplication scenario. If you have created a Management Event Incident Configuration to use for this deduplication scenario, select the Management Event Incident Configuration that you have created.
Comparison Criteria	Specify the attribute values that must match before the incident is identified as a duplicate. The possible attributes consist of the following choices. • **Name** - The **Name** attribute value from the Incident form: General tab. • **CIA** - Represents any of the following items configured as a Parameter Value

Deduplication Attributes, continued

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td></td><td>
<p>using the "Deduplication Comparison Parameters Form " on page 659:</p>
<ul>
<li>The <b>Value</b> attribute from the Incident form: Custom Attributes tab</li>
<li>An SNMP varbind Object ID</li>
<li>An SNMP varbind position number</li>
</ul>
<p>If you want to use CIA as part of your comparison criteria, see "Deduplication Comparison Parameters Form " on page 659</p>
<ul>
<li><b>SourceNode</b> - The <b>Source Node</b> attribute value from the Basics attributes listed on the Incident form. The Source Node value is the IP Address or Name of the node for which the incident was generated.

<b>Note</b>: The Source Node must be stored in the NNMi database.</li>
<li><b>Source Object</b> - The <b>Source Object</b> attribute value from the Basics attributes listed on the Incident form.

<b>Note</b>: The Source Object must be stored in the NNMi database.</li>
</ul>
<p><b>Note</b>: Each attribute value in the option you select must match before the incident is identified as a duplicate. For example, if you select <b>Name</b>, only the Incident Name value must match. If you select <b>Name SourceNode SourceObject CIA</b>, the Incident Name, Source Node, Source Object, and all Custom Incident Attribute values that you configure as a Parameter Value must match before NNMi identifies the incident as a duplicate.</p>
<p>Selecting an option that includes CIA enables you to further refine the deduplication criteria. For example, you might want to configure deduplication for incidents with CIA values that specify the same State attribute value for a particular network object.</p>
<p>For a description of each Comparison Criteria option, click here.</p>
<table>
<tr><th>Comparison Criteria</th><th>Description</th></tr>
<tr><td>Name</td><td>Value of the <b>Name</b> attribute from the Incident form: General tab must match.</td></tr>
<tr><td>Name CIA</td><td>Each of the following values must match:
<ul>
<li><b>Name</b> attribute from the Incident form: General tab</li>
<li><b>CIA</b> - Represents the Value associated with any of the following items configured as a Parameter Value using the "Deduplication Comparison Parameters Form " on page 659:
<ul><li>Name of a Custom Incident Attribute (CIA) provided by NNMi. (See the Incident form: Custom Attributes tab</li></ul></li>
</ul></td></tr>
</table>
</td></tr>
</table>

Deduplication Attributes, continued

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td></td><td>
<table>
<tr><th>Comparison Criteria</th><th>Description</th></tr>
<tr><td></td><td>- An SNMP varbind Object ID
- An SNMP varbind position number
If you want to use CIA as part of your comparison criteria, see "Deduplication Comparison Parameters Form " on page 659</td></tr>
<tr><td>Name SourceNode</td><td>**Note**: Select this option only if the Source Node is stored in the NNMi database.
Each of the following values must match:
- **Name** attribute from the Incident form: General tab
- The **Source Node** attribute value from the Basics attributes listed on the Incident form</td></tr>
<tr><td>Name SourceNode CIA</td><td>**Note**: Select this option only if the Source Node is stored in the NNMi database.
Each of the following values must match:
- **Name** attribute from the Incident form: General tab
- The **Source Node** attribute value from the Basics attributes listed on the Incident form
- **CIA** - Represents the Value associated with any of the following items configured as a Parameter Value using the "Deduplication Comparison Parameters Form " on page 659:
  - The **Value** attribute from the Incident form: Custom Attributes tab
  - An SNMP varbind Object ID
  - An SNMP varbind position number
If you want to use CIA as part of your comparison criteria, see "Deduplication Comparison Parameters Form " on page 659</td></tr>
<tr><td>Name SourceObject</td><td>**Note**: Select this option only if the Source Object is stored in the NNMi database.
Each of the following values must match:
- **Name** attribute from the Incident form: General tab</td></tr>
</table>
</td></tr>
</table>

Deduplication Attributes, continued

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td></td><td>
<table>
<tr><th>Comparison Criteria</th><th>Description</th></tr>
<tr><td></td><td>• The <b>Source Object</b> attribute value from the Basics attributes listed on the Incident form.</td></tr>
<tr><td>Name
SourceObject
CIA</td><td><b>Note</b>: Select this option only if the Source Object is stored in the NNMi database.
Each of the following values must match:
• <b>Name</b> attribute from the Incident form: General tab
• The <b>Source Object</b> attribute value from the Basics attributes listed on the Incident form
• <b>CIA</b> - Represents the Value associated with any of the following items configured as a Parameter Value using the "Deduplication Comparison Parameters Form " on page 659:
▪ The <b>Name</b> attribute from the Incident form: Custom Attributes tab
▪ An SNMP varbind Object ID
▪ An SNMP varbind position number
If you want to use CIA as part of your comparison criteria, see "Deduplication Comparison Parameters Form " on page 659</td></tr>
<tr><td>Name
SourceNode
SourceObject</td><td><b>Note</b>: Select this option only if the Source Node and Source Object are stored in the NNMi database.
Each of the following values must match:
• <b>Name</b> attribute from the Incident form: General tab
• The <b>Source Node</b> attribute value from the Basics attributes listed on the Incident form
• The <b>Source Object</b> attribute value from the Basics attributes listed on the Incident form</td></tr>
<tr><td>Name
SourceNode
SourceObject
CIA</td><td><b>Note</b>: Select this option only if the Source Node and Source Object are stored in the NNMi database.
Each of the following values must match:
• <b>Name</b> attribute from the Incident form: General tab
• The <b>Source Node</b> attribute value from the Basics</td></tr>
</table>
</td></tr>
</table>

Deduplication Attributes, continued

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td></td><td>
<table>
<tr><th>Comparison Criteria</th><th>Description</th></tr>
<tr><td></td><td>attributes listed on the Incident form
• The <b>Source Object</b> attribute value from the Basics attributes listed on the Incident form
• <b>CIA</b> - Represents the Value associated with any of the following items configured as a Parameter Value using the "Deduplication Comparison Parameters Form " on page 659:
▪ The <b>Name</b> attribute from the Incident form: Custom Attributes tab
▪ An SNMP varbind Object ID
▪ An SNMP varbind position number
If you want to use CIA as part of your comparison criteria, see "Deduplication Comparison Parameters Form " on page 659</td></tr>
</table>
</td></tr>
<tr><td>Deduplication Comparison Parameters</td><td><i>Optional</i>. If you selected a Comparison Criteria that includes CIA, you must populate one or more rows in this table. See "Deduplication Comparison Parameters Form " on page 659.</td></tr>
</table>

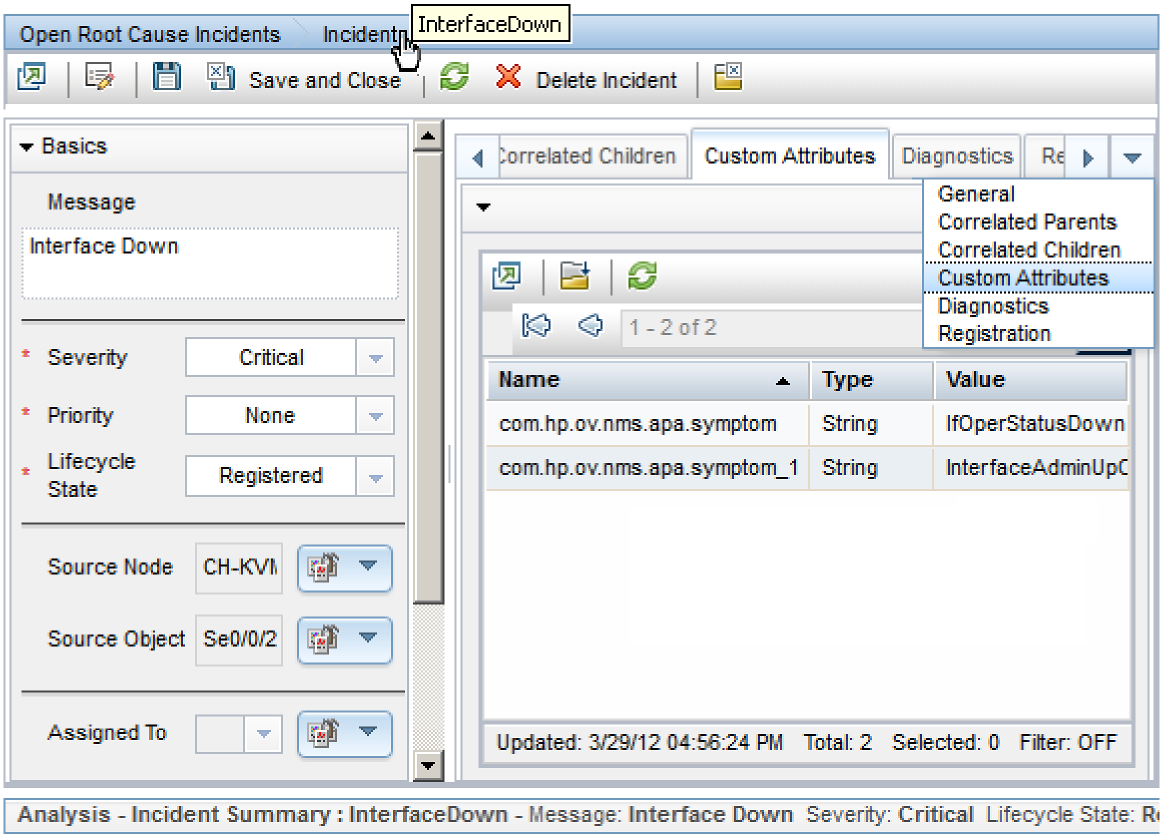
Deduplication Comparison Parameters Form (SNMP Trap Incident)

Comparison Parameter values enable accurate identification of duplicate incidents. Custom Incident Attributes (CIAs) are used as Comparison Parameter values. There are two categories of CIAs:

- SNMP trap varbind values (Name = the MIB varbind identifier, Type = asn_*)
- Custom attributes provided by NNMi (Name = cia.*, Type=String). See "Custom Incident Attributes Provided by NNMi (Information for Administrators)" on page 647.

The group of available CIAs depends on which incident you are configuring for this Deduplication (for example, CiscoLinkDown). To see which CIAs are available, navigate to an Incident view, select an instance of that incident-type, click the Open icon and navigate to the Custom Attributes tab. The items listed in the table are the CIAs for that particular incident-type. For example, all CiscoLinkDown incidents would have the same group of CIAs shown in the illustration below.

Note: You can also use the CIA (varbind) position number.



Message: Interface Down

Severity	Critical
Priority	None
Lifecycle State	Registered
Source Node	CH-KVI\
Source Object	Se0/0/2
Assigned To	

Name	Type	Value
com.hp.ov.nms.apa.symptom	String	IfOperStatusDown
com.hp.ov.nms.apa.symptom_1	String	InterfaceAdminUpC

Updated: 3/29/12 04:56:24 PM Total: 2 Selected: 0 Filter: OFF

Analysis - Incident Summary : InterfaceDown - Message: Interface Down Severity: Critical Lifecycle State: R

To specify a CIA to use in the identification criteria for duplicate incidents:

1. Navigate to the **Deduplication Comparison Params** form.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Click to expand the **Incidents** folder.
   c. Select **SNMP Trap Configurations.**
   d. Do one of the following:
      - To create a new configuration, click the ✱ New icon.
      - To edit a configuration, select a row, click the 🗁 Open icon, and continue.
   e. On the form that opens, navigate to the **Deduplication** tab.
   f. Locate the **Deduplication Comparison Parameters** table.
   g. Do one of the following to specify which CIA:
      - To add a Custom Incident Attribute parameter specification, click the ✱ New icon.
      - To edit an existing Custom Incident Attribute parameter specification, select a row, click the 🗁 Open icon, and continue.
      - To delete an existing Custom Incident Attribute parameter, select a row and click the ✖ Delete icon..
2. In the Parameter Value field, type (or copy and paste) the exact text string from the Incident

form, Custom Attribute tab, **Name** attribute value:

- NNMi-provided CIA value (see "Custom Incident Attributes Provided by NNMi (Information for Administrators)" on page 647).
- SNMP trap varbind identified by the Abstract Syntax Notation value (ASN.1).

3. Click **Save and Close** to save your changes and return to the previous configuration form.

Configure Rate (Time Period and Count) for an SNMP Trap Incident

For information about each SNMP Traps tab:

Use Rate configuration to track incident patterns *based on the number of incident reoccurrences within a specified time period*. After the count within the specified time period is reached, NNMi emits a Rate Correlation incident and continues to update the Correlation Notes with the number of occurrences within that rate.

Note: Suppression, Enrichment, and Dampening are not supported for Rate incidents.

As long as your defined criteria (Count and Hours, Minutes, Seconds) is sustained, the following information is updated in the Correlation Notes of the Rate Correlation incident:

- the actual number of occurrences of incidents for that sustained rate (Count)
- the sustained time interval (Hours, Minutes, Seconds)

For example, you can set a Rate configuration to track when a link is intermittently down at least three times in 30 minutes. NNMi shows the first occurrence of the rate incident in the incident view and uses Correlation Notes to update the number of incidents and time interval to reflect all the incremental incident occurrences and time periods. To continue the example, if the rate of three times in 30 minutes is sustained for 90 minutes, NNMi updates the Correlation Notes to specify that 9 incidents occurred in 90 minutes.

NNMi provides preconfigured Rate correlations. You can add new Rate correlations.

When you open the Incident form of the newest instance:

- On the General tab, two fields notify you that the Rate correlation is working:
  - **Correlation Nature**: Rate
  - **Count**: x
- On the **Correlated Children** tab, each incident is listed in the table.
- If a Rate Correlation Incident is dampened, note the following:
  - Rate Correlation Incidents inherit the Dampening configuration settings from its Correlated Children.
  - NNMi always retains the Parent Rate Correlation Incident, even if its Child Incidents are Closed and subsequently deleted.

    See "Dampening Incident Configurations" on page 679 for more information about Dampening an incident configuration.

To establish a rate correlation within an incident configuration:

1. Navigate to the **Rate** tab.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Expand the **Incidents** folder.
   c. Select **SNMP Trap Configurations**.
   d. Do one of the following:
      - To create a new configuration, click the ✳ New icon.
      - To edit an existing configuration, select a row, click the Open icon, and continue.
      - To delete an existing configuration, select a row and click the ✖ Delete icon.
   e. On the form that opens, locate the **Rate** tab.
2. Provide the definition for this Rate configuration (see the "Rate Configuration Definition" table).
3. *Optional*. If your Comparison Criteria includes custom incident attributes (CIA) to identify one specific incident, use the Comparison Parameter List table to define each CIA.See "Rate Comparison Parameters Form" on page 678.
4. Click **Save and Close** to save your changes and return to the previous form.

Rate Configuration Definition

Attribute	Description
Enable	Use this attribute to temporarily disable an incident's rate settings: **Enable** □ = Temporarily disable the selected configuration. **Enable** ☑ = Enable the selected configuration. If enabled, NNMi actively tracks any reoccurrences of the designated incident within the time period you specify, and generates a Rate incident.
Count	Specify the number of reoccurrences required before your Rate Configuration starts working.
Hours	Used with the Minutes and Seconds attributes to specify the time duration within which the reoccurrences are measured.
Minutes	Used with the Hours and Seconds attributes to specify the time duration within which the reoccurrences are measured.
Seconds	Used with the Hours and Minutes attributes to specify the time duration within which the reoccurrences are measured.
Parent Incident	Click the ▾ icon and select Quick Find. Select **Rate Correlation** from the list.
Comparison Criteria	Specify which group of attributes must match before the incident is identified as a duplicate. The possible groups of attributes consist of the following choices. **Name** value of the Incident (from the General tab on the Incident form).

Rate Configuration Definition , continued

Attribute	Description
	Source Node value (from the Basics group on the Incident form). Address or name of the node for which the incident was generated. **Source Object** value (from the Basics group on the Incident form). For example, the Source Object for a LinkDown incident is **interface**. **CIA** custom incident attribute values (select from the list displayed on the Custom Attributes tab on the Incident form). If you want to use CIA as part of your comparison criteria, see "Rate Comparison Parameters Form (SNMP Trap Incident)" below.
Rate Comparison Parameters	*Optional*. If you selected a Comparison Criteria that includes CIA, you must populate one or more rows in this table. See "Rate Comparison Parameters Form (SNMP Trap Incident)" below.

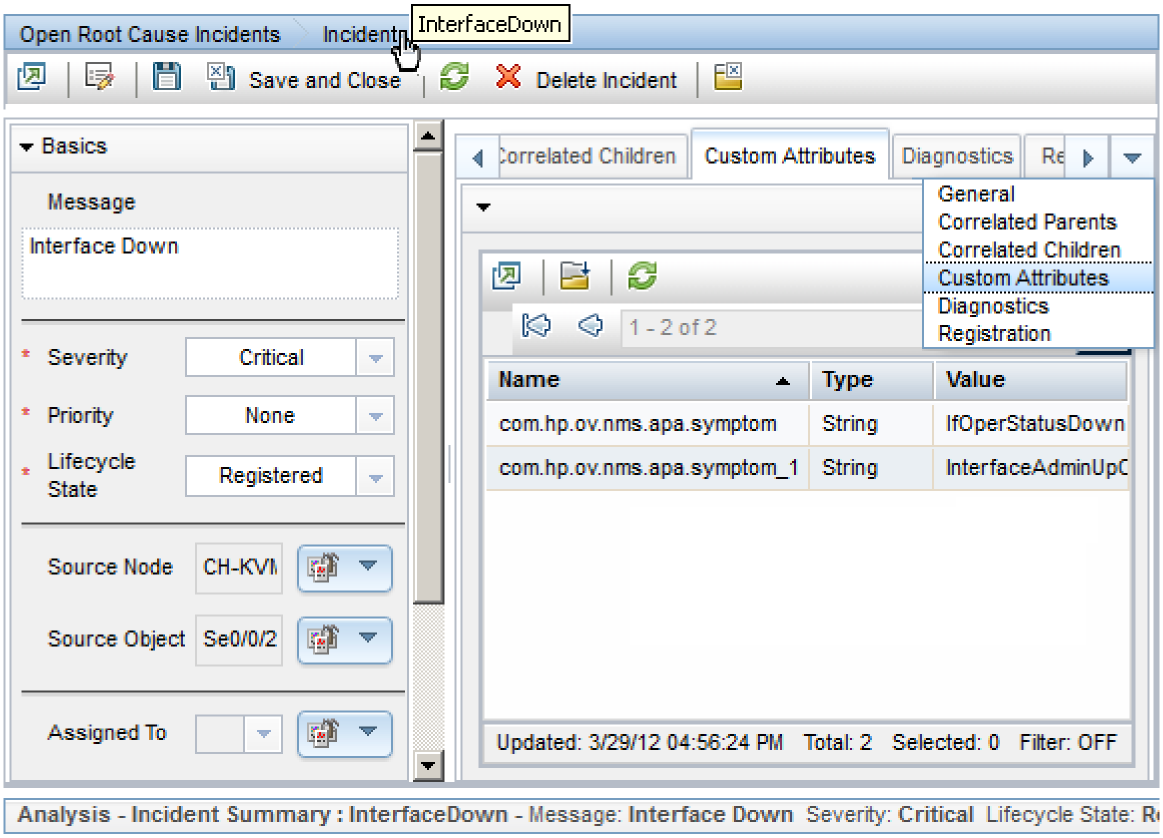
Rate Comparison Parameters Form (SNMP Trap Incident)

Custom Incident Attributes (CIAs) are used as parameter values. Parameter values enable accurate identification of duplicate incidents. There are two categories of CIAs:

- SNMP trap varbind values (Name = the MIB varbind identifier, Type = asn_*)
- Custom attributes provided by NNMi (Name = cia.*, Type=String). See "Custom Incident Attributes Provided by NNMi (Information for Administrators)" on page 647.

The group of available CIAs depends on which incident you are configuring for this Rate (for example, CiscoLinkDown). To see which CIAs are available, navigate to an Incident view, select an instance of that incident-type, click the 📂 Open icon and navigate to the Custom Attributes tab. The items listed in the table are the CIAs for that particular incident-type. For example, all CiscoLinkDown incidents would have the same group of CIAs shown in the illustration below.

Note: You can also use the CIA (varbind) position number.



Message: Interface Down

* Severity	Critical
* Priority	None
* Lifecycle State	Registered
Source Node	CH-KVI
Source Object	Se0/0/2
Assigned To	

Name	Type	Value
com.hp.ov.nms.apa.symptom	String	IfOperStatusDown
com.hp.ov.nms.apa.symptom_1	String	InterfaceAdminUpC

Updated: 3/29/12 04:56:24 PM   Total: 2   Selected: 0   Filter: OFF

Analysis - Incident Summary : InterfaceDown - Message: Interface Down Severity: Critical Lifecycle State: R

To specify a CIA to use in the identification criteria for duplicate incidents:

1. Navigate to the **Rate Comparison Params** form.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Click to expand the **Incidents** folder.
   c. Select **SNMP Trap Configurations** .
   d. Do one of the following:
      - To create a new configuration, click the New icon.
      - To edit an existing configuration, select a row, click the Open icon, and continue.
   e. On the form that opens, navigate to the **Rate** tab.
   f. Locate the **Rate Comparison Parameters** table.
   g. Do one of the following to specify which CIA:
      - To add a Custom Incident Attribute parameter specification, click the New icon.
      - To edit an existing Custom Incident Attribute parameter specification, select a row, click the Open icon, and continue.
      - To delete Custom Incident Attribute parameter specification, select a row and click the Delete icon.
2. In the Parameter Value field, type (or copy and paste) the exact text string from the Incident

form, Custom Attribute tab, **Name** attribute value:

- NNMi-provided CIA value (see "Custom Incident Attributes Provided by NNMi (Information for Administrators)" on page 647).
- SNMP trap varbind identified by the Abstract Syntax Notation value (ASN.1).

3. Click **Save and Close** to save your changes and return to the previous configuration form.

Configure Actions for an SNMP Trap Incident

For information about each SNMP Traps tab:

For information about each Actions tab:

You can configure actions to automatically run at any point in the incident lifecycle. For example, you might want to configure an action to occur when an incident of the type you are configuring is generated (**Registered**). When an incident is generated, you might want to automatically open a trouble ticket or send email or page your network operator. After the incident is **Closed**, you might want to automatically close the trouble ticket.

> **Note:** Your actions will not be executed until you enable the Actions configuration by either clicking Enable ☑ on the Actions tab or using the **Actions** → **Enable Configuration** option.

If the NNMi management server is running on a Windows operating system, NNMi runs each action that you configure using the Local System account. If the NNMi management server is running on an HP-UX, Solaris or Linux operating system, NNMi runs each action that you configure using the bin user name. To change the user account associated with actions, see the "Setting the Action Server Name Parameter" section in the *HP Network Node Manager i Software Deployment Reference*.

You can also configure incident actions based on either of the following:

- The Source Node's participation in a Node Group. See "Configure Incident Actions for a Node Group (SNMP Trap Incident)" on page 869 for more information.
- The Source Object's participation in an Interface Group. See "Configure Incident Actions for an Interface Group (SNMP Trap Incident)" on page 832 for more information.

You can configure actions for incidents generated from SNMP traps, NNM 6.x or 7.x events, and the NNMi management events. Any time an incident configuration changes, the action directory is rescanned and any executable or script files (for example, Jython) are reloaded to the NNMi database. See "Lifecycle Transition Action Form (SNMP Trap Incidents)" on the next page for more information about the actions directory.

Tip: Copy any required executable or script files to the NNMi actions directory before you configure an incident action. New or updated actions are loaded into NNMi only when an incident configuration is updated or created. See "Lifecycle Transition Action Form (SNMP Trap Incidents)" on the next page for the location of the NNMi action directory.

When the action runs, output is logged to the `incidentActions.*.*.log` file. To view the contents of the Actions log, use the **Tools** → **Incident Actions Log** menu option.

See "Verify that NNMi Services are Running" on page 85 for more information about log files and where they are located.

NNMi sets the default values described in the following table.

See the "Maintaining NNMi" chapter in the HP Network Node Manager i Software Deployment Reference for information about changing the default values for Action Server Properties.

Action Server Properties

Property	Description	Value
numProcess	Number of actions that can be run at one time.	10
numJythonThreads	Number of threads the action server uses to run Jython scripts	10
userName	User name under which the action server runs.	bin

To configure an automatic action for an incident:

1. Navigate to the **Actions** tab.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Expand the **Incidents** folder.
   c. Select **SNMP Trap Configurations**.
   d. Do one of the following:
      - To create an incident configuration, click the New icon, and continue.
      - To edit an incident configuration, select a row, click the Open icon, and continue.
      - To delete an incident configuration, select a row and click the Delete icon.
   e. Select the **Actions** tab.
2. From the **Lifecycle Actions** table toolbar, do one of the following:
   - To create an Action configuration, click the New icon, and continue.
   - To edit an Action configuration, select a row, click the Open icon, and continue.
   - To delete an Action configuration, select a row and click the Delete icon.
3. In the "Lifecycle Transition Action Form (SNMP Trap Incidents)" below, provide the required information.
4. Click **Save and Close** to save your changes and return to the previous form.

   The next time the lifecycle changes, NNMi launches the action associated with the lifecycle for the incident of that type.

Lifecycle Transition Action Form (SNMP Trap Incidents)

For information about each Action tab:

Use this form to enter the command you want to run when an incident of the type you are configuring is at a particular Lifecycle State. For example, when an incident is generated

(**Registered**), you might want to automatically open a trouble ticket or email or page your network operator.

Note: Your actions will not be executed until you enable the Actions configuration by either clicking Enable ☑ on the Actions tab or using the **Actions** → **Enable Configuration** option.

To configure an action for an incidents:

1. Navigate to the **Lifecycle Transition Actions** form:
   a. From the workspace navigation pane, select the **Configuration** workspace.
   b. Click to expand the **Incidents** folder.
   c. Select **SNMP Trap Configurations** .
   d. Select the **Actions** tab.
   e. From the **Lifecycle Transition Action** table toolbar, do one of the following:
      - To create an Action configuration, click the * New icon, and continue.
      - To edit an Action configuration, select a row, click the Open icon, and continue.
      - To delete an Action configuration, select a row and click the ✖ Delete icon.
2. Make your configuration choices (see table).

   Note: NNMi reloads the configuration information anytime the incident configuration is changed.
3. Click **Save and Close** to save your changes and return to the previous form.

Create Action Attributes

Attribute	Description
Lifecycle State	Select a Lifecycle State from the drop-down menu.
Command Type	If you provided a Jython command, select **Jython** from the drop-down list. If you are using an executable or bat file, select **ScriptOrExecutable** from the drop-down list.
Command	Enter one of the following: • A Jython method with the required parameters • Executable command for the current operating system with the required parameters. When entering a **Command** value, note the following: • Left or right bracket ([ ]) and backtick (` Unicode character: 0060 hex = 96 dec) characters are not permitted in the **Command** attribute. If you need these characters in your shell script, place them in a shell script file and reference that file from the **Command** attribute. • **Windows only**: Shell commands are not permitted in the **Command** attribute. To use shell commands, place them in a shell script file and reference that file

Create Action Attributes, continued

Attribute	Description

	from the **Command** attribute.
	• Use absolute paths to executables instead of relying on the PATH variable as it might not be set correctly.
	• Verify that you do not have two Jython methods with the same name. Otherwise, NNMi is not able to tell which is the correct method to load.
	• You can use the same Jython method for more than one incident configuration.
	• Jython (.py) files must reside in the following directory:
	Note: All the functions defined in the Jython files that reside in this directory are also accessible by NNMi. The files are also executed by NNMi on startup.
	Windows:
	`%NnmDataDir%\shared\nnm\actions`
	UNIX:
	`/var/opt/OV/shared/nnm/actions`
	• When using executable files, specify the absolute path to the executable command or make sure the directory in which the executable file resides is in your PATH environment variable.
	• NNMi provides a set of parameters that can pass attribute values from an incident configuration. See "Valid Parameters for Configuring Incident Actions (Management Events)" on page 1216 for more information.

Configure a Payload Filter for an Action (SNMP Trap Incidents)

The Payload Filter Editor enables you to create expressions that further refine the filters used to select the incidents that cause the configured action to run. Make sure to design any complex Payload Filters offline as a Boolean expression first. This method can help to minimize errors when entering your expressions using the Payload Filter editor.

To create a Payload Filter expression:

1. Navigate to the **SNMP Trap Configuration** form:
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Expand the **Incidents** folder.
   c. Select **SNMP Trap Configurations** .
   d. Do one of the following:
      i. To create an incident configuration, click the ✳ New icon, and continue.
      ii. To edit an incident configuration, select a row, click the 📂 Open icon, and continue.
      iii. To delete an incident configuration, select a row and click the ✖ Delete icon.

2. Select the **Actions** tab.

3. Do one of the following:

   a. To create a new configuration, click the ✳ New icon.

   b. To edit an existing configuration, select a row, click the Open icon, and continue.

4. Select the **Payload Filter** tab.

5. Define your Payload Filter (see table). Also see "Guidelines for Creating a Payload Filter".

   a. Plan out the logic needed for your Filter String.

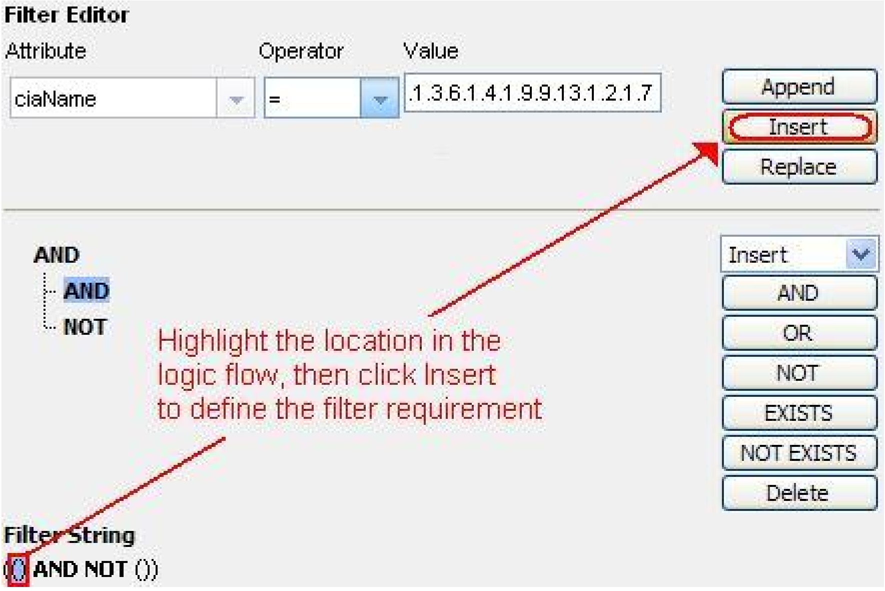
   b. Use the buttons on the bottom half of the Additional Filters Editor to establish the logic structure.

      For example, to establish the following structure, select **Insert**, then click **AND**, then **NOT**, and then **AND** a second time:

      ```
      (( ) AND NOT ( ))
      ```

   c. Now place your cursor in a location within the displayed Filter String, and use the top half of the filter editor to define the parameters of the highlighted filter requirement.

      For example, select a set of parentheses and use the Insert button to specify the filter requirement within those parentheses:



Filter Editor

Attribute: ciaName

Operator: =

Value: .1.3.6.1.4.1.9.9.13.1.2.1.7

Highlight the location in the logic flow, then click Insert to define the filter requirement

Filter String

( ) AND NOT ( )

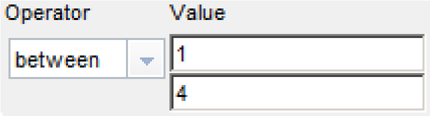
6. Click **Save and Close**.

7. Click **Save and Close** to save your changes and return to the previous form.

Payload Filter Editor Components

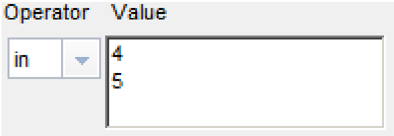
Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following:

Payload Filter Editor Components, continued

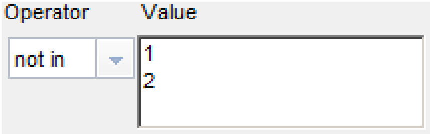
Attribute	Description
	• ciaName • ciaValue
Operator	Valid operators are described below. • **=** Finds all values equal to the value specified. Click here for an example. Example: `ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7` matches any incident that contains a varbind with the name value **.1.3.6.1.4.1.9.9.13.1.2.1.7**. • **!=** Finds all values not equal to the value specified. Click here for an example. Example: `ciaName != .1.3.6.1.4.1.9.9.13.1.2.1.7` matches any incident that contains a varbind with the name value other than **1.3.6.1.4.1.9.9.13.1.2.1.7**. • **<** Finds all values less than the value specified. Click here for an example. Example: `ciaValue < 6` matches any incident that contains a varbind value less than **6**. • **<=** Finds all values less than or equal to the value specified. Click here for an example. Example: `ciaValue <= 6` matches any incident that contains a varbind value less than or equal to **6**. • **>** Finds all values greater than the value specified. Click here for an example. Example: `ciaValue > 4` matches any incident that contains a varbind value greater than **4**. • **>=** Finds all values greater than or equal to the value specified. Click here for an example. Example: `ciaValue >= 4` matches any incident that contains a varbind with values greater than or equal to **4** . • **between** Finds all traps or events that include a varbind value equal to and between the two values specified. Click here for an example. Example: `ciaValue between`  Operator: between Value: 1 4 matches any incident that contains a varbind value equal to or greater than **1** and equal to or less than **4** . **Note**: As shown in the example, each value must be entered on a separate line. • **in** Finds any match to at least one value in a list of values. Click here for an example. Example:

Payload Filter Editor Components, continued

Attribute	Description

	`ciaValue in`
	
	Operator Value
	in 4
	5
	matches any incident that contains a varbind value of either **4** or **5**.
	Note: As shown in the example, each value must be entered on a separate line.
	NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example (**4, 5**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line.
	- **is not null** Finds all non-blank values. Click here for an example.
	Example: `ciaValue is not null` matches any incident with varbind values.
	- **is null** Finds all blank values. Click here for an example.
	Example: `ciaValue is null` matches any incident with no varbind values.
	- **like** Finds matches using wildcard characters. Click here for more information about using wildcard characters.
	The period asterisk (.*) characters mean *any number of characters of any type at this location*.
	The period (.) character means *any single character of any type at this location*.
	Note: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below.
	Examples:
	`ciaName like  \Q.1.3.6.1.4.1.9.9\E.*` finds all traps or events that contain varbind names that begin with **.1.3.6.1.4.1.9.9** and (optionally) end with any number of characters.
	`ciaValue like .*Chicago.*` finds all traps or events that contain a varbind value that includes the string **Chicago**.
	- **not between** Finds all values except those between the two values specified. Click here for an example.
	Example: `ciaValue not between 5 8` matches an incident that contains a varbind with the values less than **5** or greater than **8**.
	- **not in** Finds all values except those included in the list of values. Click here for an example.
	Example:
	`ciaValue not in`

Payload Filter Editor Components, continued

Attribute	Description
	 Operator Value not in 1 2 matches any incident that contains a varbind with values other than **1** and **2** . **Note**: As shown in the example, each value must be entered on a separate line. NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example, (**1, 2**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line. - **not like** Finds all that do not have the values specified (using wildcard strings). Click here for an example. The period asterisk (.*) characters mean *any number of characters of any type at this location*. The period (.) character means *any single character of any type at this location*. **Note**: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below. Example: `ciaName not like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that does not begin with **`.1.3.6.1.4.1.9.9`**. `ciaValue not like .*Chicago.*` finds all traps or events that do not contain a varbind value that includes the string **Chicago**.
Value	The value for which you want NNMi to search. Note the following: - The values you enter are case sensitive. - NNMi displays a variable number of value fields depending on the Operator selected. For example, the `between` Operator causes two value fields to be displayed. - The `between, in` and `not in` operators require that each value be entered on a separate line.

Additional Filters Editor Buttons

Button	Description
Append	Appends the current expression (Attribute, Operator,and Value) to the selected expression already included in the Filter String.
Insert	Inserts the current expression (Attribute, Operator,and Value) in front of the cursor location within the Filter String.

Additional Filters Editor Buttons, continued

Button	Description
Replace	Replaces the selected expression with the expression displayed in the Attribute, Operator, and Value fields.
AND	Appends, inserts, or replaces the AND Boolean Operator in the selected cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
OR	Appends, Inserts, or replaces the OR Boolean Operator in the current cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
NOT	Can be used in any part of the Filter String to specify that NNMi should exclude interfaces with values that pass the expression that immediately follows the `NOT`. For example, when evaluating the following expression, NNMi includes interfaces with (interface description) `ifDesc` containing **VLAN**, and excludes any Interfaces that have **VLAN10** for the (interface name) `ifName` value: `(ifDesc like VLAN AND NOT (ifName=VLAN10))` **Note**: View the expression displayed under Filter String to see the logic of the expression as it is created .
EXISTS	Used for filters that include Capabilities or Custom Attribute names and values in the Filer String. Indicates that you want NNMi to consider interfaces that do not have any Capabilities or Custom Attributes when evaluating the Filter String. **Note**: If you include Capabilities or Custom Attribute names and values in the Filer String, but do not use **EXISTS** or **NOT EXISTS**, NNMi excludes from its search interfaces that do not include Capabilities or Custom Attributes. For example, when evaluating the following Filter String, NNMi includes interfaces with (interface description) `ifDesc` containing **VLAN**, as well as any Interfaces Custom Attribute Role value is **LAN Connection to Oracle Server**: `(ifDesc like VLAN OR EXISTS((customAttrName=Role AND customAttrValue=LAN Connection to Oracle Server)))` **Note**: View the expression displayed under Filter String to see the logic of the expression as it is created.
NOT EXISTS	Used for filters that include Capabilities or Custom Attribute names and values in the Filer String. Indicates that you want NNMi to consider interfaces that do not have any Capabilities or Custom Attributes when evaluating the Filter String, but exclude the interfaces that match the expression that follows the **NOT EXISTS**.

Additional Filters Editor Buttons, continued

Button	Description
	Note: If you include Capabilities or Custom Attribute names and values in the Filer String, but do not use **EXISTS** or **NOT EXISTS**, NNMi excludes from its search interfaces that do not include Capabilities or Custom Attributes. For example, when evaluating the following expression, NNMi includes interfaces with (interface description) `ifDesc` containing **VLAN**, and excludes any Interfaces that have the Custom Attribute Role and that Role value is **LAN Connection to Oracle Server**: `(ifDesc like VLAN OR NOT EXISTS((customAttrName=Role AND customAttrValue=LAN Connection to Oracle Server)))` **Note**: View the expression displayed under Filter String to see the logic of the expression as it is created.
Delete	Deletes the selected expression. **Note**: If the Boolean Operator is selected, the Additional Filters Editor deletes all expressions associated with the Boolean Operator.

Valid Parameters for Configuring Incident Actions (SNMP Trap Incident)

When configuring incident actions, consider using incident information as part of the action. NNMi provides the following parameter values. Use these parameters as variables in your Jython or executable files.

Tip: See the Using the Incident Form for more information about the parameter values.

Note: NNMi stores varbind values as custom incident attributes (CIAs).

Tip: If a value is not stored for a parameter, it is returned as “null”.

See "Lifecycle Transition Action Form" on page 748 for more information about configuring incident actions.

Valid Parameters Visible From an Incident's Form

Parameter Value	Description
$category, $cat	Value of the Category attribute in the Incident form.
$count, $cnt	Value representing the number of Custom Incident Attributes that appear in the Incident form.
$family, $fam	Value from the Family attribute in the Incident form.
$firstOccurrenceTime, $fot	Value from the First Occurrence Time attribute in the incident form.
$lastOccurrenceTime, $lot	Value from the Last Occurrence Time attribute in the incident form.

Valid Parameters Visible From an Incident's Form, continued

Parameter Value	Description
$lifecycleState, $lcs	Value from the Lifecycle State attribute in the Incident form.
$name	Value of the Name attribute from the incident configuration.
$nature, $nat	Value from the Nature attribute in the Incident form.
$origin, $ori	Value from the Origin attribute in the Incident form.
$originOccurrenceTime, $oot	Value from the Origin Occurrence Time attribute in the incident form.
$priority, $pri	Value from the Priority attribute in the Incident form.
$severity, $sev	Value of the Severity attribute of the Incident form.

Valid Parameters Visible from a Node Form

Parameter Value	Description
$managementAddress, $mga	Value from the Management Address attribute of the incident's source Node's form or SNMP Agent form.
$otherSideOfConnectionManagementAddress, $oma	If the incident's Source Node is part of a Layer 2 Connection, this attribute is the value of the Management Address of a node on the other side of the Layer 2 Connection.
$sourceNodeLongName, $sln	The fully-qualified DNS name as displayed in the Hostname attribute of the incident's source Node's form.
$sourceNodeName, $snn	Value from the Name attribute of the incident's source Node's form.
$sysContact, $sct	Value from the System Contact attribute of the incident's source Node form: General tab.
$sysLocation, $slc	Value from the System Location attribute of the incident's source Node form: General tab.

Valid Parameters Visible from an Interface Form

Parameter Value	Description
$ifAlias, $ifa	Value from the IfAlias attribute for the interface that is the incident's source object.
$ifConfigDupSetting,	Configured Duplex Setting on the port associated with the interface that

Valid Parameters Visible from an Interface Form , continued

$icd	is the incident's source object.
$ifDesc, $idc	Value from the ifDesc attribute for the interface that is the incident's source object.
$ifIndex, $idx	Value from the ifIndex attribute for the interface that is the incident's source object.
$ifIpAddr, $iia	IP Address values associated with the interface that is the incident's source object. If multiple IPaddresses are associated with the interface, this parameter returns a comma-separated list.
$ifName, $ifn	Value from the ifName attribute for the interface that is the incident's source object.
$ifPhysAddr, $ipa	Value from the Physical Address attribute for the interface that is the incident's source object.
$ifSpeed, $isp	Value from the ifSpeed attribute for the interface that is the incident's souce object.
$ifType, $itp	Value from the ifType attribute for the interface that is the incident's souce object.

Valid Parameters Visible from a Layer 2 Connection Form

Parameter Value	Description
$otherSideOfConnectionConfigDupSetting, $ocd	If the incident's source Node is part of a Layer 2 Connection, this parameter contains the Configured Duplex Setting on the port associated with the interface on the other side of the connection.
$otherSideOfConnectionIfAlias, $oia	If the incident's Source Node is part of a Layer 2 Connection, this parameter is the value of the ifAlias of one of the interfaces on the other side of the Layer 2 Connection.
$otherSideOfConnectionIfDesc, $odc	If the incident's Source Node is part of a Layer 2 Connection, this parameter contains the ifDescr attribute value for the interface on the other side of the Layer 2 Connection.
$otherSideOfConnectionIfIndex, $odx	If the incident's Source Node is part of a Layer 2 Connection, this parameter contains the ifIndex attribute value for the interface on the other side of the connection.
$otherSideOfConnectionIfName, $ofn	If the incident's Source Node is part of a Layer 2 Connection, this parameter contains the ifName attribute value for the interface on the other side of the connection.

Valid Parameters Visible from a VLAN Form

Parameter Value	Description
$impVlanIds, $ivi	Value from the VLAN Id attribute associated with the interface that is the incident's source object. To access this information from an interface form, navigate to the VLAN Port tab and open the form for the VLAN of interest. If the interface is part of more than one VLAN, this parameter returns a comma-separated list.
$impVlanNames, $ivn	Value from the Global VLAN Name attribute associated with the interface that is the incident's source object. To access this information from a Node form or Interface form, navigate to the VLAN Ports tab. If the node or interface is part of more than one VLAN, this parameter returns a comma-separated list.

Valid Parameters Not Visible From a Form

Parameter Value	Description
$id	Unique Object Identifier attribute value for the incident (unique across the entire NNMi Database).
$firstOccurrenceTimeMs, $fms	Value from the First Occurrence Time attribute in the incident form, converted to millseconds (measured since January 1, 1970, 00:00:00 GMT - Greenwich Mean Time).
$lastOccurrenceTimeMs, $lms	Value from the Last Occurrence Time attribute in the incident form, converted to millseconds (measured since January 1, 1970, 00:00:00 GMT - Greenwich Mean Time).
$messageFormat, $msg	*Valid for Incident actions only*. Message text displayed for an incident when this parameter is included as an argument to an incident action.
$oid	Value of the unique object identifier (oid) for the incident configuration that originated from either an SNMP trap, Management Event, or Remote NNM 6.x or 7.x event.
$otherSideOfConnection, $osc	If the incident's Source Node is part of a Layer 2 Connection, this attribute is the following combination of values for the node and one of its interfaces on the other side of the Layer 2 Connection: The fully-qualified DNS name of the node appended with the interface Name in the following format: <*fully-qualified DNS name*>[*interface_name*]
$originOccurrenceTimeMs, $oms	Value from the Origin Occurrence Time attribute in the incident form, converted to millseconds (measured since January 1, 1970, 00:00:00 GMT - Greenwich Mean Time).

Valid Parameters Not Visible From a Form, continued

Parameter Value	Description
$sourceNodeUuid, $snu	Universally Unique Object Identifier attribute value of the source node object for the incident (unique across all databases). This identifier distinguishes the source node object instance from all other node objects.
$sourceObjectClass, $soc	Value of the object class for the object you want to include. Use this parameter to request more details of a class of objects through a web service. Examples of object classes include: `com.hp.ov.nms.model.core.Interface` and `com.hp.ov.nms.model.snmp.SnmpAgent`.
$sourceObjectName, $son	Value from the Name attribute of the source object. For example, an interface object is named according to the MIB ifName. Each ifName varies according to the vendor's conventions. Using the name `4/1` as an example, `4` represents `the board number and 1` represents the port number.
$sourceObjectUuid, $sou	Universally Unique Object Identifier attribute value of the source object for the incident (unique across all databases). This identifier distinguishes the source object instance from all other similar object instances..
$uuid	Universally Unique Object Identifier attribute value of the incident (unique across all databases). This identifier distinguishes the incident object instance from all other incident objects.

Valid Parameters Established in Custom Incident Attributes

Parameter Value	Description
$<position_ number>	Value of the custom incident attribute (CIA) position number for any CIA that originated from a varbind or was added by NNMi. For example, to indicate you want to use the varbind in position 1, enter: `$1` NNMi stores varbind values as Custom Incident Attributes. If you know the varbind position number, use this parameter.
$<CIA_ name>	Value of the name that is used for the custom incident attribute. For example, `$mycompany.mycia.` NNMi provides CIA values for configuring Management Events. See Custom Incident Attributes Provided by NNMi for more information about custom incident attributes.
$<CIA_ oid>	Value of the object identifier for any custom incident attribute that originated as a varbind. For example, `$.1.3.6.1.6.3.1.1.5.1.` Use this parameter when you are not certain of a custom incident attribute (varbind) position number.
$*	Used to indicate you want all of the custom incident attribute values originating as varbinds, to be passed to the action configuration. Each varbind is returned in the

Valid Parameters Established in Custom Incident Attributes, continued

Parameter Value	Description
	following format: `$<CIA_name>:<CIA_value>` in which the custom incident attribute name appears followed by the custom incident attribute value.

The function described in the following table replaces the specified numeric value with the associated text value stored in the CIA.

Note: The associated MIB must have been loaded using the nnmloadmib.ovpl command.

Functions to Generate Values Within Incident Messages

Function	Description
$text ($<position_number>)	The <*position_number*> argument specifies the numeric value of the custom incident attribute (CIA) position number for any CIA that originated from a varbind or was added by NNMi. For example, to indicate you want to use the varbind in position 1, enter: `$1.` After the function runs, NNMi replaces the numeric value with the text value stored in the CIA. **Note**: If a text value is not available, NNMi returns the numeric value.
$text ($<CIA_oid>)	The <CIA_oid> argument specifies the object identifier for any custom incident attribute that originated as a varbind. For example, `$.1.3.6.1.6.3.1.1.5.1.` Use this argument to the $text function when you are not certain of a custom incident attribute (varbind) position number. After the function runs, NNMi replaces the numeric value with the text value stored in the CIA. **Note**: If a text value is not available, NNMi returns the numeric value.

Configure Forward to Global Manager Settings for an SNMP Trap Incident (*NNMi Advanced*)

For information about each SNMP Traps tab:

(*NNMi Advanced - Global Network Management feature*) The NNMi Global Network Management feature enables multiple NNMi management servers to work together while managing different geographic areas of your network. See NNMi's Global Network Management Feature (NNMi Advanced) for more information. The Global Manager combines topology information from multiple Regional Managers, but maintains *a separate set of incidents about those nodes*.

Use the Global Manager Forwarding tab when you want to forward specific SNMP traps from your NNMi management server (a Regional Manager) to all Global Managers in your Global Network Management environment.

> **Caution:** The Global Manager must have an incident configuration for that SNMP trap, otherwise the incoming trap is dropped. See "Export and Import Configuration Settings" on page 1579 for ideas about sharing incident configurations among NNMi management servers.

When you configure Forward to Global Managers, you can specify an optional Payload Filter for NNMi to use when determining *which occurrences* should be forwarded to Global Managers. Payload Filters enable you to use the data that is included with an occurrence of an incident configuration before it is stored as an incident in the NNMi database.

Examples of the type of data that can be used as a Payload Filter include Custom Incident Attribute names (ciaName) and values (ciaValue). For example, you might want NNMi to forward an incident based on a particular status change notification trap. To do so, you would specify a Payload Filter that includes the name of the Custom Incident Attribute that stores the status information as well as the status change value string of interest.

> **Tip:** See also "Configure Trap Forwarding Destinations" on page 1381.

To configure Forwarding to Global Managers:

1. Navigate to the **SNMP Trap Configuration** form:
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Expand the **Incidents** folder.
   c. Select **SNMP Trap Configurations** .
   d. Do one of the following:
      i. To create an incident configuration, click the New icon, and continue.
      ii. To edit an incident configuration, select a row, click the Open icon, and continue.
      iii. To delete an incident configuration, select a row and click the Delete icon.
2. Select the **Forward to Global Managers** tab.
3. Provide the required information (see table)
4. Click **Save and Close** to save your changes and return to the previous form.

Forwarding Configuration Attributes

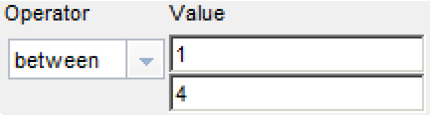
Name	Description
Enable	Use this attribute to enable or temporarily disable an incident's Forward to Global Managers settings: **Enable** □ = Temporarily disable the selected configuration. **Enable** ☑ = Enable the selected configuration.
Payload Filter	The Payload Filter Editor enables you to create expressions that further refine the filters used to select the incidents that NNMi forwards to other servers. Make sure to design any complex Payload Filters offline as a Boolean expression first. This method can help to minimize errors when entering your expressions using the Payload Filter

Forwarding Configuration Attributes , continued

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td></td><td>
<p>editor.</p>
<p>When creating a Payload Filter, note the following:</p>
<ul>
<li>Payload Filter expressions for the <code>like</code> and <code>not like</code> operators use the syntax defined for java regular expressions (java.util.regex Pattern class).</li>
<li>You must use a <code>ciaName</code> that already exists in the trap or event you are configuring.</li>
<li>Each set of expressions associated with a Boolean Operator is treated as if it were enclosed in parentheses and evaluated together.</li>
<li>View the expression displayed under <b>Filter String</b> to see the logic of the expression as it is created.</li>
<li>The <code>AND</code> and <code>OR</code> Boolean Operators must contain at least two expressions as shown in the example below.
<p>The following example filters incidents on voltage state:</p>
<pre>AND
      ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7
      ciaValue = 5</pre>
<p>NNMi evaluates the expression above as follows:</p>
<pre>(ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7 AND ciaValue = 5)</pre>
<p>NNMi finds all incidents with a varbind value of <code>.1.3.6.1.4.1.9.9.13.1.2.1.7</code> and CIA value of <b>5</b>.</p></li>
<li>The placement of your cursor and the subsequent text that is selected is important when performing operations using the Payload Filter Editor. For example, you append to, replace, or change the indentation of the expression that is selected.</li>
<li>The placement of your cursor and the subsequent text that is selected is especially important when adding your Boolean operators.</li>
</ul>
<p><b>Payload Filter Editor Components</b></p>
<table>
<tr><th>Attribute</th><th>Description</th></tr>
<tr><td>Attribute</td><td>The attribute name on which NNMi searches. Filterable attributes include the following:<ul><li>ciaName</li><li>ciaValue</li></ul></td></tr>
<tr><td>Operator</td><td>Valid operators are described below.<ul><li><b>=</b> Finds all values equal to the value specified. Click here for an example.</li></ul></td></tr>
</table>
</td></tr>
</table>

Forwarding Configuration Attributes , continued

Name	Description
	Payload Filter Editor Components, continued

Attribute	Description
	Example: `ciaName=.1.3.6.1.4.1.9.9.13.1.2.1.7` matches any incident that contains a varbind with the name value **.1.3.6.1.4.1.9.9.13.1.2.1.7**. - **!=** Finds all values not equal to the value specified. Click here for an example. Example: `ciaName! = .1.3.6.1.4.1.9.9.13.1.2.1.7` matches any incident that contains a varbind with the name value other than **1.3.6.1.4.1.9.9.13.1.2.1.7**. - **<** Finds all values less than the value specified. Click here for an example. Example: `ciaValue < 6` matches any incident that with a varbind value less than **6**. - **<=** Finds all values less than or equal to the value specified. Click here for an example. Example: `ciaValue <= 6` matches any incident that with a varbind value less than or equal to **6**. - **>** Finds all values greater than the value specified. Click here for an example. Example: `ciaValue > 4` matches any incident that with a varbind value greater than **4**. - **>=** Finds all values greater than or equal to the value specified. Click here for an example. Example: `ciaValue >= 4` matches any incident that contains a varbind with values greater than or equal to **4**. - **between** Finds all values equal to and between the two values specified. Click here for an example. Example: `ciaValue between`  Operator: between Value: 1 4 matches any incident that contains a varbind value equal to or greater than **1** and equal to or less than **4**. **Note:** As shown in the example, each value must be entered on a separate line.

Forwarding Configuration Attributes , continued

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td></td><td>

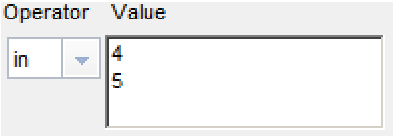
Payload Filter Editor Components, continued

<table>
<tr><th>Attribute</th><th>Description</th></tr>
<tr><td></td><td>

- **in** Finds any match to at least one value in a list of values. Click here for an example.

  Example:

  `ciaValue in`

  Operator: in
  Value:
  4
  5

  matches any incident that with a varbind value of either **4** or **5**.

  > **Note:** As shown in the example, each value must be entered on a separate line.

  NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example (**4, 5**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line.

- **is not null** Finds all non-blank values. Click here for an example.

  Example: `ciaValue is not null` matches any incident with a varbind value.

- **is null** Finds all blank values. Click here for an example.

  Example: `ciaValue is null` matches any incident with no varbind values.

- **like** Finds matches using wildcard characters. Click here for more information about using wildcard characters.

  The period asterisk (.*) characters mean *any number of characters of any type at this location*.

  The period (.) character means *any single character of any type at this location*.

  > **Note:** To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below.

  Example:

  `ciaName like \Q.1.3.6.1.4.1.9.9\E.*` matches any

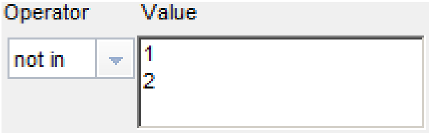
</td></tr>
</table>

</td></tr>
</table>

Forwarding Configuration Attributes , continued

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td></td><td>

Payload Filter Editor Components, continued

<table>
<tr><th>Attribute</th><th>Description</th></tr>
<tr><td></td><td>incident that contains a varbind name value that begins with 1.3.6.1.4.1.9.9 and (optionally) ends with any number of characters.
`ciaValue like .*Chicago.*` finds all traps or events that contain a varbind value that includes the string **Chicago**.
• **not between** Finds all values except those between the two values specified. Click here for an example.
Example: `ciaValue not between 5 8` matches an incident that contains a varbind with the values less than **5** or greater than **8** .
• **not in** Finds all values except those included in the list of values. Click here for an example.
Example:
`ciaValue not in`
Operator: not in
Value:
1
2
matches any incident that contains a varbind with values other than **1** and **2**.
Note: As shown in the example, each value must be entered on a separate line.
NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example, (**1, 2**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line.
• **not like** Finds all that do not have the values specified (using wildcard strings). Click here for an example.
The period asterisk (.*) characters mean *any number of characters of any type at this location*.
The period (.) character means *any single character of any type at this location*.
Note: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below.</td></tr>
</table>

</td></tr>
</table>

Forwarding Configuration Attributes , continued

Name	Description

Payload Filter Editor Components, continued

Attribute	Description
	Example: `ciaName not like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that does not begin with **.1.3.6.1.4.1.9.9** and (optionally) ends with any number of characters. `ciaValue not like .*Chicago.*` finds all traps or events that do not contain a varbind value that includes the string **Chicago**.
Value	The value for which you want NNMi to search. Note the following: • The values you enter are case sensitive. • NNMi displays a variable number of value fields depending on the Operator selected. For example, the `between` Operator causes two value fields to be displayed. • The `between, in` and `not in` operators require that each value be entered on a separate line.

Payload Filter Editor Buttons

Button	Description
Append	Appends the current expression (Attribute, Operator,and Value) to the selected expression already included in the filter string.
Insert	Inserts the current expression (Attribute, Operator,and Value) in front of the cursor location within the Filter String.
Replace	Replaces the selected expression with the expression displayed in the Attribute, Operator, and Value fields.
AND	Inserts the AND Boolean Operator in the selected cursor location. **Note:** View the expression displayed under **Filter String** to see the logic of the expression as it is created.
OR	Inserts the OR Boolean Operator in the current cursor location. **Note:** View the expression displayed under **Filter String** to see the logic of the expression as it is created.
NOT	Can be used in any part of the Filter String to specify that NNMi

Forwarding Configuration Attributes , continued

Name	Description

Payload Filter Editor Buttons, continued

Button	Description
	should exclude interfaces with values that pass the expression that immediately follows the `NOT`. For example, when evaluating the following expression, NNMi includes interfaces with (interface description) `ifDesc` containing **VLAN**, and excludes any Interfaces that have **VLAN10** for the (interface name) `ifName` value: `(ifDesc like VLAN AND NOT (ifName=VLAN10))` **Note:** View the expression displayed under Filter String to see the logic of the expression as it is created.
EXISTS	Used for filters that include Capabilities or Custom Attribute names and values in the Filer String. Indicates that you want NNMi to consider interfaces that do not have any Capabilities or Custom Attributes when evaluating the Filter String. **Note:** If you include Capabilities or Custom Attribute names and values in the Filer String, but do not use **EXISTS** or **NOT EXISTS**, NNMi excludes from its search interfaces that do not include Capabilities or Custom Attributes. For example, when evaluating the following Filter String, NNMi includes interfaces with (interface description) `ifDesc` containing **VLAN**, as well as any Interfaces Custom Attribute Role value is **LAN Connection to Oracle Server**: `(ifDesc like VLAN OR EXISTS((customAttrName=Role AND customAttrValue=LAN Connection to Oracle Server)))` **Note:** View the expression displayed under Filter String to see the logic of the expression as it is created.
NOT EXISTS	Used for filters that include Capabilities or Custom Attribute names and values in the Filer String. Indicates that you want NNMi to consider interfaces that do not have any Capabilities or Custom Attributes when evaluating the Filter String, but exclude the interfaces that match the expression that follows the **NOT EXISTS**.

Forwarding Configuration Attributes , continued

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td></td><td>
<b>Payload Filter Editor Buttons, continued</b>
<table>
<tr><th>Button</th><th>Description</th></tr>
<tr><td></td><td>
<b>Note:</b> Note: If you include Capabilities or Custom Attribute names and values in the Filer String, but do not use <b>EXISTS</b> or <b>NOT EXISTS</b>, NNMi excludes from its search interfaces that do not include Capabilities or Custom Attributes.
For example, when evaluating the following expression, NNMi includes interfaces with (interface description) <code>ifDesc</code> containing <b>VLAN</b>, and excludes any Interfaces that have the Custom Attribute Role and that Role value is <b>LAN Connection to Oracle Server</b>:
<code>(ifDesc like VLAN OR NOT EXISTS ((customAttrName=Role AND customAttrValue=LAN Connection to Oracle Server)))</code>
<b>Note:</b> View the expression displayed under Filter String to see the logic of the expression as it is created.
</td></tr>
<tr><td>Delete</td><td>
Deletes the selected expression.
<b>Note:</b> If the Boolean Operator is selected, the Payload Filter Editor deletes all expressions associated with the Boolean Operator.
</td></tr>
</table>
</td></tr>
</table>

Configure Syslog Message Incidents (HP ArcSight)

The HP NNMi–ArcSight integration adds syslog message information to NNMi, so that NNMi users can view these syslog messages and investigate potential problems. After the ArcSight integration is enabled, NNMi receives `ArcSightEvent` traps that contain syslog message data. NNMi then maps this syslog information to a Syslog Message incident configuration and treats it as a syslog message in NNMi. See the *HP Network Node Manager i Software-HP ArcSight Logger Integration Guide*for more information.

You can configure how you want these incidents to be displayed in the incident views provided by NNMi. The types of things you configure include name, category, and the message format.

Note: When the Source Object for a Syslog Message Incident is a Port object, NNMi resolves the Source Object to the associated Interface. Because ArcSight does not store Interface data, these incidents do not appear in the ArcSight user interface. See the HP Network Node

> Manager i Software-HP ArcSight Logger Integration Guidefor more information about best practices for viewing these incidents.

To configure a Syslog Message incident:

1. Navigate to the **Syslog Message Configuration** form.
    a. From the workspace navigation panel, select the **Configuration** workspace.
    b. Expand the **Incidents** folder.
    c. Select **Syslog Message Configurations**.
2. Do one of the following:
    a. To create a Syslog Message incident configuration, click the * New icon, and continue.
    b. To edit a Syslog Message incident configuration, double-click the row representing the configuration you want to edit, and continue.
    c. To delete a Syslog Message configuration, select a row, and click the ✖ Delete icon.
3. In the Syslog Message Configuration form, provide the required information.
4. Click **Save and Close** to save your changes and return to the **Incident Configuration** form.

The next time that a syslog message event of this type arrives into the database, NNMi creates an associated incident to display in the appropriate console incident views.

Syslog Message Configuration Form (HP ArcSight)

To configure incidents originating from syslog messages:

1. Navigate to the **Syslog Message Configuration** form:
    a. From the workspace navigation pane, select the **Configuration** workspace.
    b. Expand the **Incidents** folder.
    c. Select **Syslog Message Configurations**.
2. Make your configuration choices (see table).

    Note: If you want to add or edit a Syslog Message incident configuration, verify that **Enabled** ☑ is selected.

    a. To add a Syslog Message incident configuration, click the * New icon, and continue.
    b. To edit a Syslog Message incident configuration, double-click the row representing the configuration you want to edit, and continue.
    c. To delete a Syslog Message incident configuration, click the ✖ Delete icon.
3. Click **Save and Close** to save your changes and return to the previous form.

Tasks for Syslog Message Incident Configuration

Task	How
"Specify the Incident Configuration Name (Syslog Messages) (HP ArcSight)" on page 941	Use the **Basics** group of the **Syslog Message Configuration** form. Specify a name that helps you to identify the configuration for subsequent use.
Specify whether you want to enable this configuration.	In the **Basics** group of the **Syslog Message Configuration** form, verify that **Enable** ☑ is selected for each configuration you want to use.
"Specify Category and Family Attribute Values for Organizing Your Incidents (Syslog Message) (HP ArcSight)" on page 941	Use the **Basics** group of the **Syslog Message Configuration** form. You can organize your incidents using Category and Family.
"Specify the Incident Severity (Syslog Message) (HP ArcSight)" on page 946	Use the **Basics** group of the **Syslog Message Configuration** form. Possible Severity values include: **Normal, Warning, Minor, Major,** and **Critical**.
"Specify Your Incident Message Format (Syslog Message) (HP ArcSight)" on page 946	Use the **Basics** group of the **Syslog Message Configuration** form. The message format determines the message to be displayed for the incident.
"Specify a Description for Your Incident Configuration (Syslog Messages)(HP ArcSight)" on page 954	Use the **Basics** group of the **Syslog Message Configuration** form. Provide a meaningful description.
Specify an Author for Your Incident Configuration (Management Events)	Use the **Basics** pane of the **Syslog Message Configuration** form to indicate who created or last modified the event. **Caution:** If the Author attribute value is **HP Network Node Manager**, any changes are at risk of being overwritten in the future. • Click **Lookup** and select **Show Analysis** to display details about the currently selected Author. • Click **Quick Find** to access the list of existing Author values. • Click **New** to create an Author value.

After you complete the Basic Configuration for the Syslog Message incident, you can also choose to configure the information described in the following table.

Additional Configurations

Task	How
"Correlate Duplicate Incidents (Deduplication Configuration)" on page 659	Select the **Deduplication** tab to specify duplicate incidents that you want to be suppressed.
"Track Incident Frequency (Rate: Time Period and Count)" on page 659	Select the **Rate** tab to specify a rate for duplicate incidents. After the rate limit is reached, NNMi generates an Incident to notify you of the problem.
"Configure an Action for an Incident" on page 748	Select the **Actions** tab to specify actions that should occur automatically when an incident changes its Lifecycle State.
"Configure Diagnostics for an Incident (NNM iSPI NET)" on page 757	Select the **Node Settings** tab to specify diagnostic actions that should occur automatically when an incident reaches a selected Lifecycle State for a node that belongs to a particular Node Group.

Configure Basic Settings for a Syslog Message Incident (HP ArcSight)

The Basics settings for a Syslog Message incident specifies general information for an incident configuration, including the name, severity, and message.

Note: In the **Basics** group of the **Syslog Message Configuration** form, verify that **Enable** ☑ is selected for each configuration you want to use.

For information about each Syslog Messages tab:

To configure Basic settings for a Syslog Message incident:

Navigate to the **Syslog Message Configuration** form:

1. From the workspace navigation panel, select the **Configuration** workspace.
2. Expand the **Incidents** folder.
3. Select **Syslog Message Configurations**.
4. Do one of the following:
   a. To create an incident configuration, click the ✳ New icon, and continue.
   b. To edit an incident configuration, select a row, click the Open icon, and continue.
   c. To delete an incident configuration, select a row, and click the ✖ Delete icon.
5. Configure the required Basic settings (see the Basic Attributes table).
6. Click **Save and Close** to save your changes and return to the previous form. NNMi uses the SNMP Object ID to enable forwarding of Management Events as SNMP traps. NNMi automatically assigns a unique SNMP Object ID to all Management Events provided by NNMi.

Basic Attributes for Syslog Message Configuration

Task	How
"Specify the Incident Configuration Name (Syslog Messages) (HP ArcSight)" on page 941	Use the **Basics** pane of the **Syslog Message Configuration** form. Specify the value of the `AdditionalDataValue` mnemonic for the undefined trap as the Syslog Message name. In the following example `LINK-3-UPDOWN` is the `AdditionalDataValue` mnemonic value for the trap: `additionalDataValue.1 .1.3.6.1.4.1.11937.1.42.1.3.1 LINK-3-UPDOWN` Alpha-numeric, spaces, and the following special characters are permitted: - (dash), _ (underscore), : (colon), and / (slash). If the mnemonic value includes non-supported characters, replace each character with an underscore character (_) or space. See the *HP Network Node Manager i Software-HP ArcSight Logger Integration Guide*.for more information.
Specify whether you want to enable this configuration.	In the **Basics** group of the **Syslog Message Configuration** form, verify that **Enable** ☑ is selected for each configuration you want to use.
"Specify Category and Family Attribute Values for Organizing Your Incidents (Syslog Message) (HP ArcSight)" on page 941	Use the **Basics** pane of the **Syslog Message Configuration** form. You can organize your incidents using Category and Family.
"Specify the Incident Severity (Syslog Message) (HP ArcSight)" on page 946	Use the **Basics** pane of the **Syslog Message Configuration** form. Possible Severity values include: **Normal, Warning, Minor, Major,** and **Critical**.
"Specify Your Incident Message Format (Syslog Message) (HP ArcSight)" on page 946	Use the **Basics** pane of the **Syslog Message Configuration**form. The message format determines the message to be displayed for the incident.
"Specify a Description for Your Incident Configuration (Syslog Messages)(HP ArcSight)" on page 954	Use the **Basics** pane of the **Syslog Message Configuration** form. Provide a meaningful description.
Specify an Author for Your Incident	Use the **Basics** pane of the **Syslog Message**

Basic Attributes for Syslog Message Configuration, continued

Task	How
Configuration (Management Events)	**Configuration** form to indicate who created or last modified the event. **Caution:** If the Author attribute value is **HP Network Node Manager**, any changes are at risk of being overwritten in the future. • Click **Lookup** and select **Show Analysis** to display details about the currently selected Author. • Click **Quick Find** to access the list of existing Author values. • Click **New** to create an Author value.

After you complete the Basic Configuration for the remote NNM 6.x/7.x event, you can also choose to configure the information described in the following table.

Additional Incident Configurations

Task	How
"Configure Interface Settings for a Syslog Message Incident (HP ArcSight)" on page 954	Select the **Interface Settings** tab to specify an Interface Group to which you want your incident configuration to apply.
"Configure Node Settings for a Syslog Message Incident (HP ArcSight)" on page 991	Select the **Node Settings** tab to specify a Node Group to which you want your incident configuration to apply.
"Configure Suppression Settings for a Syslog Message Incident (HP ArcSight)" on page 1030	Select the **Suppression** tab to specify the criteria for discarding incidents that match the selected incident configuration.
"Configure Enrichment Settings for a Syslog Message Incident (HP ArcSight)" on page 1038	Select the **Enrichment** tab to specify enhancements for the selected incident configuration.
"Configure Dampening Settings for a Syslog Message Incident (HP ArcSight)" on page 1042	Select the **Dampen** tab to specify the time interval that must be met before the incident appears in an Incident view.
"Configure Deduplication for a Syslog Message Incident (HP ArcSight)" on page 1051	Select the **Deduplication** tab to specify duplicate incidents that you want to be suppressed.
"Configure Rate (Time Period and Count) for a Syslog Message Incident (HP ArcSight)" on page 1059	Select the **Rate** tab to specify a rate for duplicate incidents. After the rate limit is reached, NNMi generates an Incident to notify you of the problem.

Additional Incident Configurations, continued

Task	How
"Configure Actions for a Syslog Message Incident (HP ArcSight)" on page 1063	Select the **Actions** tab to specify actions that should occur automatically when an incident changes its Lifecycle State.

Specify the Incident Configuration Name (Syslog Messages) (HP ArcSight)

Specify the value of the `AdditionalDataValue` mnemonic as the Syslog Message name.

In the following example `LINK-3-UPDOWN` is the `AdditionalDataValue` mnemonic value for the trap:

```
additionalDataValue.1 .1.3.6.1.4.1.11937.1.42.1.3.1 LINK-3-UPDOWN
```

Valid characters include alphanumeric, dash (-), slash (/), colon (:), and underscore(_).

See the HP Network Node Manager i Software-HP ArcSight Logger Integration Guide for more information.

Specify Category and Family Attribute Values for Organizing Your Incidents (Syslog Message) (HP ArcSight)

When configuring incidents, NNMi provides the Category and Family attributes to help you organize your incidents.

Preconfigured Categories
The Category attribute helps you organize your incidents. Select the category that you want to be associated with this type of incident when it appears in an incident view. Each of the possible Category values is described in the following table.

Incident Categories Provided by NNMi

Category	Description
Accounting	Used to indicate problems related to usage statistics and allocation of costs associated with the billing of time and services provided by devices. This category is not used by NNMi with default configurations, but it is available for incidents you define.
Application Status	Indicates there is a problem with the health of the NNMi software. Examples of these kinds of events include license expiration (see "Extend a Licensed Capacity" on page 1575) or that a certain NNMi process or service lost connection to the Process Status Manager (see "Stop or Start an NNMi Process" on page 82 and "Stop or Start NNMi Services" on page 86).
Configuration	Indicates there is a problem with the configuration of a managed device. For example, there is a physical address mismatch.

Incident Categories Provided by NNMi, continued

Category	Description
Fault	Indicates a problem with the network, for example Node Down.
Performance	Indicates a Monitored Attribute value *crossed* a configured threshold. For example, Disk Space Utilization exceeds the configured threshold criteria for High Value = 90 percent .
Security	Indicates there is a problem related to authentication. For example, an SNMP authentication failure.
Status	Indicates some kind of status message. Examples of these kinds of incidents include "SNMP Link Up" or an "HSRP Group status Normal" message.

Note: You can add your own Category entries to NNMi. See "Create an Incident Category (Management Events)" on page 1087 for more information.

You can use **Family** attribute values to further categorize the types of incidents that might be generated. Each of the possible values are described in the following table.

Incident Family Attribute Values Provided by NNMi

Family	Description
Address	Indicates the incident is related to an address problem.
Aggregated Port	Indicates the incident is related to a **Link Aggregation**[1] problem.
BGP	Indicates the incident is related to a problem with BGP (Border Gateway Protocol). This family is not used by NNMi with default configurations, but it is available for incidents you define.
Board	Indicates the incident is related to a board problem. This family is not used by NNMi with default configurations, but it is available for incidents you define.
Chassis	Indicates the incident is related to an board problem. This family is not used by NNMi with default configurations, but it is available for incidents you define.
Component Health	Indicates the incident is related to Node Component metrics collected by NNMi. See "Node Form: Node Component Tab" for more information about the Node Component metrics collected.
Connection	Indicates the incident is related to a problem with one or more connections.

[1] Protocols used on Switches to configure multiple Interfaces (Aggregation Member Interfaces) to function as if they were one (an Aggregator Interface). When two Aggregator Interfaces establish a connection, that connection is an Aggregator Layer 2 Connection. The Aggregator Layer 2 Connection appears on Layer 2 Neighbor View maps as a thick line with an Interface icon at each end (representing the Aggregator Interface). When you double-click the thick line, it converts into multiple thin lines representing the participating Aggregation Member Layer 2 Connections with their Aggregation Member Interfaces at each end of the lines.

Incident Family Attribute Values Provided by NNMi, continued

Family	Description
Correlation	Indicates the incident has additional incidents correlated beneath it. These incidents are associated with a duplicate count so that you can determine the number of correlated incidents associated with it.
Custom Poller	Indicates the incident is related to the NNMi Custom Poller feature. See "About Custom Poller".
HSRP	*NNMi Advanced*. Indicates the incident is related to a problem with Hot Standby Router Protocol (**HSRP**[1]).
Interface	Indicates the incident is related to a problem with one or more interfaces.
License	Indicates the incident is related to a licensing problem.
NNMi Health	Indicates the incident is related to NNMi Health. See the Check NNMi Health for more information.
Node	Indicates the incident is related to a node problem.
OSPF	Indicates the incident is related to an OSPF problem. This family is not used by NNMi with default configurations, but it is available for incidents you define.
RAMS	*NNMi Advanced*. Indicates the incident is related to a Router Analytics Management System problem.
RMON	Indicates the incident is related to a Remote Monitor (IETF standard, RFC 1757) problem. This family is not used by NNMi with default configurations, but it is available for incidents you define.
RRP	*NNMi Advanced*. Indicates the incident is related to a problem with a Router Redundancy Protocol configuration.
STP	Indicates the incident is related to Spanning-Tree Protocol problem. This family is not used by NNMi with default configurations, but it is available for incidents you define.
Syslog	NNMi does not use this Family with default configurations. It is available for incidents you define.
Trap Analysis	Indicates the incident is related to an SNMP trap storm.
VLAN	Indicates the incident is related to a problem with a virtual local area network.
VRRP	*NNMi Advanced*. Indicates the incident is related to a problem with Virtual Router Redundancy Protocol (**VRRP**[2]).

[1] Hot Standby Router Protocol
[2] Virtual Router Redundancy Protocol

Note: You can add your own Family entries to NNMi. See "Create an Incident Family (Syslog Message) (HP ArcSight)" on the next page for more information.

Create an Incident Category (Syslog Message) (HP ArcSight)

The Category attribute helps you organize your incidents. Create any Category that makes sense to you and your team. For a list of the Category codes provided by NNMi, "Specify Category and Family Attribute Values for Organizing Your Incidents (Syslog Message) (HP ArcSight)" on page 941.

To create a new incident Category:

1. Navigate to the **Incident Category** form.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Expand the **Incidents** folder.
   c. Select **Syslog Message Configurations**.
   d. Do one of the following:
      - To create an incident configuration, click the New icon, and continue.
      - To edit an incident configuration, select a row, click the Open icon, and continue.
      - To delete an incident configuration, select a row, and click the Delete icon.
   e. In the configuration form, locate the **Category** attribute.
   f. Click the Lookup icon, and select New.
2. Provide the required information (see table).
3. Click **Save and Close** to save your changes and return to the previous form.

Category Code Attributes

Name	Description
Label	Incident category name. For example, Hardware Faults, or Cisco Error. Maximum size is 255 characters. Alpha-numeric, spaces, and underline characters are permitted.

Unique Key	**Caution**: After you click **Save and Close**, this value cannot be changed.
	Used as a unique identifier when exporting and importing configuration definitions. To ensure that the value you enter is unique, it is recommended that you use the Java name space convention when providing this value. It is also useful to include the label value as part of the unique key as shown in the following examples:
	`com.<your_company_name>.nnm.trapConf.category.<category_label>`
	`com.<your_company_name>.nnm.eventConf.category.<category_label>`
	`com.<your_company_name>.nnm.inciConf.category.<category_label>`
	The maximum length is 80 characters. Alpha-numeric characters and periods are permitted. Spaces are not permitted.

Create an Incident Family (Syslog Message) (HP ArcSight)

The Family attribute helps you organize your incidents. Create any Family that makes sense to you and your team. For a list of the Family codes provided by NNMi, "Specify Category and Family Attribute Values for Organizing Your Incidents (Syslog Message) (HP ArcSight)" on page 941.

To create a new incident Family:

1. Navigate to the **Incident Family** form.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Expand the **Incidents** folder.
   c. Select **Syslog Message Configurations** .
   d. Do one of the following:
      - To create an incident configuration, click the ✱ New icon, and continue.
      - To edit an incident configuration, select a row, click the Open icon, and continue.
      - To delete an incident configuration, select a row, and click the ✖ Delete icon.
   e. In the configuration form, locate the **Family** attribute.
   f. Click the Lookup icon, and select ✱ New.
2. Provide the required information (see table).
3. Click **Save and Close** to save your changes and return to the previous form.

Family Attributes

Name	Description
Label	Family name. For example, Hardware Faults, or Cisco Error. Maximum size is 255 characters. Any character type is valid.

Unique Key	**Caution**: After you click **Save and Close**, this value cannot be changed.
	Used as a unique identifier when exporting and importing configuration definitions. To ensure that the value you enter is unique, it is recommended that you use the Java name space convention when providing this value. It is also useful to include the label value as part of the unique key as shown in the following examples:
	`com.<your_company_name>.nnm.trapConf.family.<family_label>`
	`com.<your_company_name>.nnm.eventConf.family.<family_label>`
	`com.<your_company_name>.nnm.inciConf.family.<family_label>`
	The maximum length is 80 alpha-numeric characters, periods allowed, no spaces allowed.

Specify the Incident Severity (Syslog Message) (HP ArcSight)

The incident severity represents the seriousness calculated for the incident. Use the severity attribute to specify the severity that should be assigned to the incident you are configuring. Possible values are described in the following table.

Incident Severity Values

Attribute	Description
Normal	Indicates there are no known problems related to the associated object. This severity is meant to be informational. Generally, no action is needed for these incidents.
Warning	Indicates there might be a problem related to the associated object.
Minor	Indicates NNMi has detected problems related to the associated object that require further investigation.
Major	Indicates NNMi has detected problems related to the associated object to be resolved before they become critical.
Critical	Indicates NNMi has detected problems related to the associated object that require immediate attention.

See "Monitor Incidents for Problems" for more information about these severity values.

Specify Your Incident Message Format (Syslog Message) (HP ArcSight)

When configuring an incident, specify the information you want NNMi to include in the incident's Message attribute value. You can use any combination of valid parameter strings and Custom Incident attributes to configure the Message.

Note: The incident Message limit is 1024 characters. If the returned values exceed this limit, NNMi truncates the value starting from the end of the returned text string.

"Valid Parameters for Configuring Incident Messages (Syslog Message) (HP ArcSight)" below

"Include Custom Incident Attributes in Your Message Format (Syslog Message) (HP ArcSight)" on page 953

Valid Parameters for Configuring Incident Messages (Syslog Message) (HP ArcSight)

When configuring incident messages, consider using incident information as part of the message. NNMi provides the following parameter values. Use these parameters as variables when formatting an incident message.

Tip: See the Using the Incident Form for more information about the parameter values.

Note: NNMi stores varbind values as custom incident attributes (CIAs).

Tip: If a value is not stored for a parameter, it is returned as “null”.

See "Specify Your Incident Message Format (Syslog Message) (HP ArcSight)" on the previous page for more information about configuring messages.

Parameter strings are available for the following:

Note: See the following tables to view the valid parameters for incidents generated from Custom Polled Instances: Parameter Strings for all Incidents (Attributes from an Incident form), Parameter Strings for Node Source Objects (Attributes from a Node form), and the Parameter Strings for all Incidents (Attributes not Visible from any form).

- Parameter strings for all incidents (Incident form attributes) (Click here for a list of choices.)

Parameter Strings for all Incidents (Incident form attributes)

Parameter String	Description
$category, $cat	Value of the Category attribute in the Incident form.
$count, $cnt	Value representing the number of Custom Incident Attributes that appear in the Incident form.
$family, $fam	Value from the Family attribute in the Incident form.
$firstOccurrenceTime, $fot	Value from the First Occurrence Time attribute in the incident form.
$lastOccurrenceTime, $lot	Value from the Last Occurrence Time attribute in the incident form.
$lifecycleState, $lcs	Value from the Lifecycle State attribute in the Incident form.
$name	Value of the Name attribute from the incident configuration.
$nature, $nat	Value from the Nature attribute in the Incident form.
$origin, $ori	Value from the Origin attribute in the Incident form.
$originOccurrenceTime, $oot	Value from the Origin Occurrence Time attribute in the incident form.
$priority, $pri	Value from the Priority attribute in the Incident form.
$sev, $severity	Value of the Severity attribute of the Incident form.

- Parameter Strings for Node Source Objects (Node form attributes) (Click here for a list of choices.)

Parameter Strings for Node Source Objects (Node form attributes)

Parameter String	Description
$managementAddress, $mga	Value from the Management Address attribute of the incident's source Node's form or SNMP Agent form.

Parameter Strings for Node Source Objects (Node form attributes) , continued

Parameter String	Description
$otherSideOfConnectionManagementAddress, $oma	If the incident's Source Node is part of a Layer 2 Connection, this attribute is the value of the Management Address of a node on the other side of the Layer 2 Connection.
$sourceNodeLongName, $sln	The fully-qualified DNS name as displayed in the Hostname attribute of the incident's source Node's form.
$sourceNodeName, $snn	Value from the Name attribute of the incident's source Node's form.
$sysContact, $sct	Value from the System Contact attribute of the incident's source Node form: General tab.
$sysLocation, $slc	Value from the System Location attribute of the incident's source Node form: General tab.

- Parameter Strings for Interface Source Objects (Interface form attributes) (Click here for a list of choices.)

Parameter Strings for Interface Source Objects (Interface form attributes)

Parameter String	Description
$ifAlias, $ifa	Value from the IfAlias attribute for the interface that is the incident's source object.
$ifConfigDupSetting, $icd	Configured Duplex Setting on the port associated with the interface that is the incident's source object.
$ifDesc, $idc	Value from the ifDesc attribute for the interface that is the incident's source object.
$ifIndex, $idx	Value from the ifIndex attribute for the interface that is the incident's source object.
$ifIpAddr, $iia	IP Address values associated with the interface that is the incident's source object. If multiple IPaddresses are associated with the interface, this parameter returns a comma-separated list.
$ifName, $ifn	Value from the ifName attribute for the interface that is the incident's source object.
$ifPhysAddr, $ipa	Value from the Physical Address attribute for the interface that is the incident's source object.
$ifSpeed, $isp	Value from the ifSpeed attribute for the interface that is the incident's

Parameter Strings for Interface Source Objects (Interface form attributes) , continued

Parameter String	Description
	souce object.
$ifType, $itp	Value from the ifType attribute for the interface that is the incident's souce object.

- Parameter Strings for Layer 2 Connection Source Objects (Layer 2 Connection form attributes) (Click here for a list of choices.)

Parameter Strings for Layer 2 Connection Source Objects (Layer 2 Connection form attributes)

Parameter String	Description
$otherSideOfConnectionConfigDupSetting, $ocd	If the incident's source Node is part of a Layer 2 Connection, this parameter contains the Configured Duplex Setting on the port associated with the interface on the other side of the connection.
$otherSideOfConnectionIfAlias, $oia	If the incident's Source Node is part of a Layer 2 Connection, this parameter is the value of the ifAlias of one of the interfaces on the other side of the Layer 2 Connection.
$otherSideOfConnectionIfDesc, $odc	If the incident's Source Node is part of a Layer 2 Connection, this parameter contains the ifDescr attribute value for the interface on the other side of the Layer 2 Connection.
$otherSideOfConnectionIfIndex, $odx	If the incident's Source Node is part of a Layer 2 Connection, this parameter contains the ifIndex attribute value for the interface on the other side of the connection.
$otherSideOfConnectionIfName, $ofn	If the incident's Source Node is part of a Layer 2 Connection, this parameter contains the ifName attribute value for the interface on the other side of the connection.

- Parameter strings for VLAN Source Objects (VLAN form attributes) (Click here for a list of choices.)

Parameter Strings for VLAN Source Objects (VLAN form attributes)

Parameter String	Description
$impVlanIds, $ivi	Value from the VLAN Id attribute associated with the interface that is the incident's source object. To access this information from an interface

Parameter Strings for VLAN Source Objects (VLAN form attributes), continued

Parameter String	Description
	form, navigate to the VLAN Port tab and open the form for the VLAN of interest. If the interface is part of more than one VLAN, this parameter returns a comma-separated list.
$impVlanNames, $ivn	Value from the VLAN Name attribute associated with the interface that is the incident's source object. To access this information from an interface form, navigate to the VLAN Ports tab of the Interface form. If the interface is part of more than one VLAN, this parameter returns a comma-separated list.

- Parameter Strings for all incidents (Additional information that is not visible in any form) (Click here for a list of choices.)

Parameter Strings for all Incidents (Attributes not visible in any form)

Parameter String	Description
$firstOccurrenceTimeMs, $fms	Value from the First Occurrence Time attribute in the incident form, converted to millseconds (measured since January 1, 1970, 00:00:00 GMT - Greenwich Mean Time).
$lastOccurrenceTimeMs, $lms	Value from the Last Occurrence Time attribute in the incident form, converted to millseconds (measured since January 1, 1970, 00:00:00 GMT - Greenwich Mean Time).
$oid	Value of the unique object identifier (oid) for the incident configuration that originated from either an SNMP trap, Management Event, or Remote NNM 6.x or 7.x event.
$otherSideOfConnection, $osc	If the incident's Source Node is part of a Layer 2 Connection, this attribute is the following combination of values for the node and one of its interfaces on the other side of the Layer 2 Connection: The fully-qualified DNS name of the node appended with the interface Name in the following format: <*fully-qualified DNS name*>[*interface_name*]
$originOccurrenceTimeMs $oms	Value from the Origin Occurrence Time attribute in the incident form, converted to millseconds (measured since January 1, 1970, 00:00:00 GMT - Greenwich Mean Time).
$sourceNodeUuid, $snu	Universally Unique Object Identifier attribute value of the source node object for the incident (unique across all databases). This identifier distinguishes the source node object instance from all other node objects.
$sourceObjectClass,	Value of the object class for the object you want to include. Use

Parameter Strings for all Incidents (Attributes not visible in any form), continued

Parameter String	Description
$soc	this parameter to request more details of a class of objects through a web service. Examples of object classes include: `com.hp.ov.nms.model.core.Interface` and `com.hp.ov.nms.model.snmp.SnmpAgent`.
$sourceObjectName, $son	Value from the Name attribute of the source object. For example, an interface object is named according to the MIB ifName. Each ifName varies according to the vendor's conventions. Using the name `4/1` as an example, `4` represents `the board number and 1` represents the port number.
$sourceObjectUuid, $sou	Universally Unique Object Identifier attribute value of the source object for the incident (unique across all databases). This identifier distinguishes the source object instance from all other similar object instances.
$uuid	Universally Unique Object Identifier attribute value of the incident (unique across all databases). This identifier distinguishes the incident object instance from all other incident objects.

- Information established in Custom Incident Attributes (Click here for a list of choices.)

Parameter Strings for Attributes Established in Custom Incident Attributes

Parameter String	Description
$<position _number>	Value of the custom incident attribute (CIA) position number for any CIA that originated from a varbind or was added by NNMi. For example, to indicate you want to use the varbind in position 1, enter: `$1` NNMi stores varbind values as Custom Incident Attributes. If you know the varbind position number, use this parameter.
$<CIA_ name>	Value of the name that is used for the custom incident attribute. For example, `$mycompany.mycia`. NNMi provides CIA values for configuring Management Events. See Custom Incident Attributes Provided by NNMi for more information about custom incident attributes.
$<CIA_ oid>	Value of the object identifier for any custom incident attribute that originated as a varbind. For example, `$.1.3.6.1.6.3.1.1.5.1`. Use this parameter when you are not certain of a custom incident attribute (varbind) position number.
$*	Used to indicate you want all of the custom incident attribute values originating as varbinds, to be passed to the action configuration. Each varbind is returned in the following format: `$<CIA_name>:<CIA_value>` in which the custom incident attribute name appears followed by the custom incident attribute value.

- Functions to generate values (Click here for a list of choices.)

The function described in the following table replaces the specified numeric value with the associated text value stored in the CIA.

Note: The associated MIB must have been loaded using the nnmloadmib.ovpl command.

Functions to Generate Values Within the Incident Message

Function	Description
$oidtext ($<position_number>)	A <*position_number*> argument specifies the numeric value of the custom incident attribute (CIA) position number for any CIA that originated from a varbind or was added by NNMi. For example, `$oidtext($2)`. **Note**: The position number you enter must represent a CIA that contains an Object Identifier (OID) value. NNMi returns the textual value of the OID for the CIA specified. Note the following: ▪ If the MIB is not loaded, NNMi returns the numeric OID value. ▪ If the OID has a MIB instance, the number representing the MIB instance is appended to the textual OID value.
$oidtext ($<CIA_oid>)	The <*CIA_oid*> argument specifies the Object Identifier (OID) for any custom incident attribute that originated as a varbind. For example, `$oidtext ($.1.3.6.1.6.3.1.1.5.1.)` Use this argument to the $oidtext() function when you are not certain of a custom incident attribute (varbind) position number. NNMi replaces the numeric value with the textual value of the OID you specify. Note the following: ▪ If the MIB is not loaded, NNMi returns the numeric OID value. ▪ If the OID has a MIB instance, the number representing the MIB instance is appended to the textual OID value.
$text ($<position_number>)	The <*position_number*> argument specifies the numeric value of the custom incident attribute (CIA) position number for any CIA that originated from a varbind or was added by NNMi. For example, to indicate you want to use the varbind in position 1, enter: `$1`. NNMi replaces the numeric value with the text value stored in the CIA. **Note**: If a text value is not available, NNMi returns the numeric value.
$text ($<CIA_oid>)	The <CIA_oid> argument specifies the object identifier for any custom incident attribute that originated as a varbind. For example, `$.1.3.6.1.6.3.1.1.5.1`. Use this argument to the $text function when you are not certain of a custom incident attribute (varbind) position number. NNMi replaces the numeric value with the text value stored in the CIA. **Note**: If a text value is not available, NNMi returns the numeric value.

Include Custom Incident Attributes in Your Message Format (Syslog Message) (HP ArcSight)

NNMi includes two categories of CIAs:

- SNMP trap varbinds identified by the Abstract Syntax Notation value (ASN.1). NNMi turns varbinds into CIAs and maintains each varbind's position number. See "Load SNMP Trap Incident Configurations" on page 771.
- Custom incident attributes provided by NNMi. See "Custom Incident Attributes Provided by NNMi (Information for Administrators)" on page 647.

  You cannot create Custom Incident Attributes.

You can use CIAs in your message format to extend the amount of information presented. To determine which CIAs are available for any particular incident type, open an Incident view, locate the incident and open the Incident form. Navigate to the **Custom Attributes** tab. A complete list of available CIAs (for that incident type) appears in the table.

To include a CIA in your message format, type the dollar-sign character (`$`) plus any of the following:

- Varbind position number or asterisk (*) to include all varbind values
- Name of the CIA
- Object identifier (oid) of the CIA (useful when the varbind position number is not consistent among vendors)

Note: A single incident cannot include two CIAs with the same name. However, two incidents can contain CIAs having the same names and values.

The following table presents some example formats with the subsequent output.

Example Incident Message Formats

Example Message Format	Output in Incident View
Possible trouble with $3	`Possible trouble with` <varbind 3>
Possible trouble with $11	`Possible trouble with` <varbind 11>
Possible trouble with $77 (where the varbind position 77 does not exist)	`Possible trouble with <Invalid or unknown cia> 77`
Possible trouble with $*	`Possible trouble with` <cia1_name: cia_value>, <cia2_name; cia_value>,< cia*n*_name: cia_value>
Possible trouble with $3x	`Possible trouble with` <varbind 3>`x`
Possible trouble with $1.2.3.4.5	`Possible trouble with` <value of the CIA with oid of 1.2.3.4.5>
Possible trouble with $mycia.mycompany	`Possible trouble with` <value of the CIA with name of mycia.mycompany>

Tip: NNMi provides an error message when a CIA cannot be found. For example, if you enter an unavailable varbind position, name, or object identifier (oid), NNMi returns an "Invalid or unknown cia" error message.

Specify a Description for Your Incident Configuration (Syslog Messages)(HP ArcSight)

NNMi provides the Description attribute to help you further identify the current incident configuration.

Description

Use the description field to provide additional information that you would like to store about the current incident configuration. This description applies only to the configuration entry.

Type a maximum of 1024 characters. Alpha-numeric, spaces, and special characters (~ ! @ # $ % ^ & * ( ) _+ -) are permitted.

Configure Interface Settings for a Syslog Message Incident (HP ArcSight)

Note: Interface Settings override any other Suppression, Enrichment, Dampen, or Actions settings for this incident, including those configured on the Node Settings tab.

NNMi enables you to apply an incident configuration to a Source Object based on the Source Object's participation in an Interface Group. If the Source Object is not a member of the Interface Group specified, the incident is neither displayed nor stored in the NNMi database

Tip: See "Create Interface Groups" on page 321 for more information about Interface Groups.

For information about each Interface Settings tab:

For information about each Syslog Message tab:

To apply an incident configuration to a Source Object based on the Source Object's Interface Group:

1. Navigate to the **Syslog Message Configuration** form:
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Expand the **Incidents** folder.
   c. Select **Syslog Message Configurations**.
   d. Do one of the following:
      i. To create an incident configuration, click the New icon, and continue.
      ii. To edit an incident configuration, select a row, click the Open icon, and continue.
      iii. To delete an incident configuration, select a row, and click the Delete icon.
2. Select the **Interface Settings** tab.

3. Do one of the following:
   a. To create a new configuration, click the New icon.
   b. To edit an existing configuration, double-click the row representing the configuration you want to edit.
4. Configure the desired Interface Settings (see table).
5. Configure any Suppression, Dampen, or Enrichment settings for this Interface Group.
6. Click **Save and Close** to save your changes and return to the previous form.

Interface Group Attributes

Name	Description
Interface Group	Click the Lookup icon and select Quick Find to select the Interface Group you want to use. See "Use the Quick Find Window" on page 41 for more information about using Quick Find.
Ordering	Determines the priority order for those interfaces that appear in multiple Interface Groups. The lower the number, the higher the priority. For example, **1** is the highest priority. If an interface is in multiple Interface Groups and more than one of those Interface Groups have been specified in an incident configuration, only the incident configuration with the highest priority will be applied to the interface.
Enable	Use this attribute to temporarily disable an incident's configuration settings: **Enable** □ = Temporarily disable the selected configuration. **Enable** ☑ = Enable the selected configuration.

Related Topics

"Configure Node Settings for a Syslog Message Incident (HP ArcSight)" on page 991

Configure Incident Suppression Settings for an Interface Group (Syslog Message)(HP ArcSight)

Note: Interface Settings override any other Suppression settings for this incident, including those from the Node Settings tab.

NNMi enables you to suppress a specified incident configuration based on the Source Object's participation in an Interface Group.

Note: You can also suppress the incident configuration based on the Source Node's participation in a Node Group. See "Configure Incident Suppression Settings for a Node Group (Syslog Message) (HP ArcSight)" on page 992 for more information.

Tip: See "Create Interface Groups" on page 321 for more information about Interface Groups.

For information about each Interface Settings tab:

To suppress an incident configuration based on an Interface Group:

1. Navigate to the **Syslog Message Configuration** form:
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Expand the **Incidents** folder.
   c. Select **Syslog Message Configurations**.
   d. Do one of the following:
      i. To create an incident configuration, click the ✱ New icon, and continue.
      ii. To edit an incident configuration, select a row, click the Open icon, and continue.
      iii. To delete an incident configuration, select a row, and click the ✖ Delete icon.
2. Select the **Interface Settings** tab.
3. Do one of the following:
   a. To create a new configuration, click the ✱ New icon.
   b. To edit a configuration, double-click the row representing the configuration you want to edit.
4. Make sure you configure the basic Interface Setting behavior. See "Configure Interface Settings for a Syslog Message Incident (HP ArcSight)" on page 954 for more information.
5. Select the **Suppression** tab.
6. Configure the desired Suppression behavior (see table).
7. Click **Save and Close** to save your changes and return to the previous form.

Interface Settings Suppression Attributes

Name	Description
Enable	Use this attribute to temporarily disable an incident's suppression settings for the specified Interface Group: **Enable** □ = Temporarily disable the selected configuration. **Enable** ☑ = Enable the selected configuration.
Payload Filter	The Payload Filter Editor enables you to create expressions that further refine the filters used to select the incidents to be suppressed, enriched, or dampened. Make sure to design any complex Payload Filters offline as a Boolean expression first. This method can help to minimize errors when entering your expressions using the Payload Filter editor. When creating a Payload Filter, note the following: • Payload Filter expressions for the `like` and `not like` operators use the syntax defined for java regular expressions (java.util.regex Pattern class). • You must use a `ciaName` that already exists in the trap or event you are configuring. • Each set of expressions associated with a Boolean Operator is treated as if it were enclosed in parentheses and evaluated together.

Interface Settings Suppression Attributes , continued

Name	Description

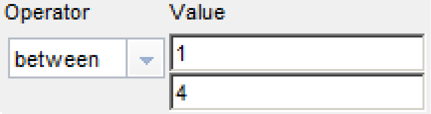
	• View the expression displayed under **Filter String** to see the logic of the expression as it is created.
	• The `AND` and `OR` Boolean Operators must contain at least two expressions as shown in the example below.
	The following example filters incidents on voltage state:
	`AND`
	`    ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7`
	`    ciaValue = 5`
	NNMi evaluates the expression above as follows:
	`(ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7 AND ciaValue = 5)`
	NNMi finds all incidents with a varbind `.1.3.6.1.4.1.9.9.13.1.2.1.7` value of **5**.
	• The placement of your cursor and the subsequent text that is selected is important when performing operations using the Payload Filter Editor. For example, you append to, replace, or change the indentation of the expression that is selected.
	• The placement of your cursor and the subsequent text that is selected is especially important when adding your Boolean operators.
	• You can include more than one varbind in the same Payload Filter expression as shown in the following example:
	`((ciaName like \Q.1.3.6.1.4.1.9.9\E.* AND ciaValue = 25) AND (ciaName like \Q.1.3.6.1.2.1.2.2.1.1.3\E.* AND ciaValue = 3))`
	In this example, a given trap must meet each of the following criteria:
	▪ Contain a varbind whose Object Identifier (OID) matches the regular expression `\Q.1.3.6.1.4.1.9.9\E.*` and has a value of `25`.
	▪ Contain a varbind whose OID matches the regular expression `\Q.1.3.6.1.2.1.2.2.1.1.3\E.*` and has a value of `3`.

Payload Filter Editor Components

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: • ciaName • ciaValue
Opera	Valid operators are described below.

Interface Settings Suppression Attributes , continued

Name	Description
	Payload Filter Editor Components, continued

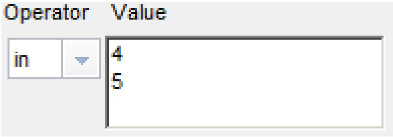
Attribute	Description
tor	• **=** Finds all values equal to the value specified. Click here for an example. Example: `ciaName=.1.3.6.1.4.1.9.9.13.1.2.1.7` matches any incident that contains a varbind with the name value of **.1.3.6.1.4.1.9.9.13.1.2.1.7**. • **!=** Finds all values not equal to the value specified. Click here for an example. Example: `ciaName! = .1.3.6.1.4.1.9.9.13.1.2.1.7` matches any incident that contains a varbind with a name value other than **1.3.6.1.4.1.9.9.13.1.2.1.7**. • **<** Finds all values less than the value specified. Click here for an example. Example: `ciaValue < 6` matches any incident that contains a varbind with a value less than **6**. • **<=** Finds all values less than or equal to the value specified. Click here for an example. Example: `ciaValue <= 6` matches any incident that contains a varbind with a value less than or equal to **6**. • **>** Finds all values greater than the value specified. Click here for an example. Example: `ciaValue > 4` matches any incident that contains a varbind with a value greater than **4**. • **>=** Finds all values greater than or equal to the value specified. Click here for an example. Example: `ciaValue >= 4` matches any incident that contains a varbind with values greater than or equal to **4**. • **between** Finds all values equal to and between the two values specified. Click here for an example. Example: `ciaValue between`  Operator: between Value: 1 4 matches any incident that contains a varbind value equal to or greater than **1** and equal to or less than **4**. **Note**: As shown in the example, each value must be entered on a separate line.

Interface Settings Suppression Attributes , continued

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td></td><td>

Payload Filter Editor Components, continued

<table>
<tr><th>Attribute</th><th>Description</th></tr>
<tr><td></td><td>

- **in** Finds any match to at least one value in a list of values. Click here for an example.
Example:
`ciaValue in`
Operator Value
in
4
5
matches any incident with a varbind value of either **4** or **5**.
Note: As shown in the example, each value must be entered on a separate line.
NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example (**4, 5**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line.
- **is not null** Finds all non-blank values. Click here for an example.
Example: `ciaValue is not null` matches any incident with a varbind that contains a value.
- **is null** Finds all blank values. Click here for an example.
Example: `ciaValue is null` matches any incident with a varbind that does not contain a value.
- **like** Finds matches using the syntax defined for java regular expressions. See the Pattern (Java Platform SE6) API documentation at : `http://download.oracle.com/javase/6/docs/api/java/util/regex/Pattern.html` for more information. Click here for more information.
The period asterisk (.*) characters mean *any number of characters of any type at this location*.
The period (.) character means *any single character of any type at this location*.
Note: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below.

</td></tr>
</table>

</td></tr>
</table>

Interface Settings Suppression Attributes , continued

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td></td><td>

Payload Filter Editor Components, continued

<table>
<tr><th>Attribute</th><th>Description</th></tr>
<tr><td></td><td>

Example:

`ciaName like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that begins with 1.3.6.1.4.1.9.9 and (optionally) ends with any number of characters.

`ciaValue like .*Chicago.*` finds all traps or events that contain a varbind value that includes the string **Chicago**.

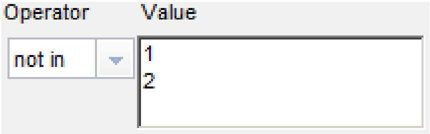
- **not between** Finds all values except those between the two values specified. Click here for an example.

  Example: `ciaValue not between 5 8` matches an incident that contains a varbind with the values less than **5** or greater than **8** .

- **not in** Finds all values except those included in the list of values. Click here for an example.

  Example:

  `ciaValue not in`

  not in

  1
  2

  matches any incident that contains a varbind with values other than **1** and **2**.

  Note: As shown in the example, each value must be entered on a separate line.

  NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example, (**1, 2**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line.

- **not like** Finds all that do not have the values specified using the syntax defined for Java regular expressions. See the Pattern (Java Platform SE6) API documentation at: `http://download.oracle.com/javase/6/docs/api/java/util/regex/Pattern.html` for more information. Click here for an example.

  The period asterisk (.*) characters mean *any number of characters of any type at this location*.

</td></tr>
</table>

</td></tr>
</table>

Interface Settings Suppression Attributes , continued

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td></td><td>

Payload Filter Editor Components, continued

<table>
<tr><th>Attribute</th><th>Description</th></tr>
<tr><td></td><td>

The period (.) character means *any single character of any type at this location*.

Note: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below.

Example:

`ciaName not like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that does not begin with **.1.3.6.1.4.1.9.9** and (optionally) ends with any number of characters.

`ciaValue not like .*Chicago.*` finds all traps or events that do not contain a varbind value that includes the string **Chicago**.

</td></tr>
<tr><td>Value</td><td>

The value for which you want NNMi to search.

Note the following:

- The values you enter are case sensitive.
- NNMi displays a variable number of value fields depending on the Operator selected. For example, the `between` Operator causes two value fields to be displayed.
- The `between, in` and `not in` operators require that each value be entered on a separate line.

</td></tr>
</table>

Payload Filter Editor Buttons

<table>
<tr><th>Button</th><th>Description</th></tr>
<tr><td>Append</td><td>Appends the current expression (Attribute, Operator,and Value) to the selected expression already included in the filter string.</td></tr>
<tr><td>Insert</td><td>Inserts the current expression (Attribute, Operator,and Value) in front of the cursor location within the Filter String.</td></tr>
<tr><td>Replace</td><td>Replaces the selected expression with the expression displayed in the Attribute, Operator, and Value fields.</td></tr>
<tr><td>AND</td><td>

Inserts the AND Boolean Operator in the selected cursor location.

Note: View the expression displayed under **Filter String** to see the logic of the expression as it is created.

</td></tr>
</table>

</td></tr>
</table>

Interface Settings Suppression Attributes , continued

Name	Description
	Payload Filter Editor Buttons, continued

Button	Description
OR	Inserts the OR Boolean Operator in the current cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
NOT	Can be used in any part of the Filter String to specify that NNMi should exclude interfaces with values that pass the expression that immediately follows the `NOT`. For example, when evaluating the following expression, NNMi includes interfaces with (interface description) `ifDesc` containing **VLAN**, and excludes any Interfaces that have **VLAN10** for the (interface name) `ifName` value: `(ifDesc like VLAN AND NOT (ifName=VLAN10))` **Note**: View the expression displayed under Filter String to see the logic of the expression as it is created .
EXISTS	Used for filters that include Capabilities or Custom Attribute names and values in the Filer String. Indicates that you want NNMi to consider interfaces that do not have any Capabilities or Custom Attributes when evaluating the Filter String. **Note**: If you include Capabilities or Custom Attribute names and values in the Filer String, but do not use **EXISTS** or **NOT EXISTS**, NNMi excludes from its search interfaces that do not include Capabilities or Custom Attributes. For example, when evaluating the following Filter String, NNMi includes interfaces with (interface description) `ifDesc` containing **VLAN**, as well as any Interfaces Custom Attribute Role value is **LAN Connection to Oracle Server**: `(ifDesc like VLAN OR EXISTS((customAttrName=Role AND customAttrValue=LAN Connection to Oracle Server)))` **Note**: View the expression displayed under Filter String to see the logic of the expression as it is created.
NOT EXISTS	Used for filters that include Capabilities or Custom Attribute names and values in the Filer String. Indicates that you want NNMi to consider interfaces that do not have any Capabilities or Custom

Interface Settings Suppression Attributes , continued

<table>
<tr><th>Nam
e</th><th>Description</th></tr>
<tr><td></td><td>

Payload Filter Editor Buttons, continued

<table>
<tr><th>Button</th><th>Description</th></tr>
<tr><td></td><td>Attributes when evaluating the Filter String, but exclude the interfaces that match the expression that follows the **NOT EXISTS**.
Note: If you include Capabilities or Custom Attribute names and values in the Filer String, but do not use **EXISTS** or **NOT EXISTS**, NNMi excludes from its search interfaces that do not include Capabilities or Custom Attributes.
For example, when evaluating the following expression, NNMi includes interfaces with (interface description) `ifDesc` containing **VLAN**, and excludes any Interfaces that have the Custom Attribute Role and that Role value is **LAN Connection to Oracle Server**:
`(ifDesc like VLAN OR NOT EXISTS ((customAttrName=Role AND customAttrValue=LAN Connection to Oracle Server))`
Note: View the expression displayed under Filter String to see the logic of the expression as it is created.</td></tr>
<tr><td>Delete</td><td>Deletes the selected expression.
Note: If the Boolean Operator is selected, the Payload Filter Editor deletes all expressions associated with the Boolean Operator.</td></tr>
</table>

</td></tr>
</table>

Configure Incident Enrichment Settings for an Interface Group (Syslog Message)(HP ArcSight)

Note: Interface Settings override any other Enrichment settings for this incident, including those from the Node Settings tab.

NNMi enables you to fine tune and enhance a specified incident configuration based on the Source Object's participation in an Interface Group. The types of items you can fine tune and enhance for a selected incident configuration, include:

- Category
- Family
- Severity
- Priority
- Correlation Nature

- Message
- Assigned To

Note: You can also enhance the incident configuration based on the Source Node's participation in a Node Group. See "Configure Incident Enrichment Settings for a Node Group (Syslog Message) (HP ArcSight)" on page 1000 for more information.

Tip: See Create Interface Groups for more information about Interface Groups.

For information about each Interface Settings tab:

For information about each Enrichment tab:

To enrich an incident configuration based on an Interface Group:

1. Navigate to the **Syslog Message Configuration** form:
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Expand the **Incidents** folder.
   c. Select **Syslog Message Configurations**.
   d. Do one of the following:
      i. To create an incident configuration, click the New icon, and continue.
      ii. To edit an incident configuration, select a row, click the Open icon, and continue.
      iii. To delete an incident configuration, select a row, and click the Delete icon.
2. Select the **Interface Settings** tab.
3. Do one of the following:
   a. To create a new configuration, click the New icon.
   b. To edit an existing configuration, double-click the row representing the configuration you want to edit.
4. Make sure you configure the basic Interface Setting behavior. See "Configure Interface Settings for a Syslog Message Incident (HP ArcSight)" on page 954 for more information.
5. Select the **Enrichment** tab.
6. Do one of the following:
   a. To create an Enrichment configuration, click the New icon and continue.
   b. To edit an Enrichment configuration, select a row, click the Open icon, and continue.
   c. To delete an Enrichment configuration, select a row and click the Delete icon.
7. Configure the desired Enrichment behavior (see the "Enrich Configuration Attributes" table)
8. Click **Save and Close** to save your changes and return to the previous form.

Interface Settings Enrichment Configuration Attributes

Name	Description
Category	Use the Category attribute to customize the category for this incident configuration. Possible values include: • Accounting • Application Status • Configuration • Fault • Performance • Security • Status See "Specify Category and Family Attribute Values for Organizing Your Incidents (Syslog Message) (HP ArcSight)" on page 941 for more information.
Family	Use the Family attribute to customize the Family for this incident configuration. Select from the drop-down list or create a new value. For example, some of the values provided by NNMi include: • Address • Aggregated Port (Interfaces using **Link Aggregation**[1] protocol. See Interface Form: Link Aggregation tab.) • Card • Connection • Correlation • Interface • Node See "Specify Category and Family Attribute Values for Organizing Your Incidents (Syslog Message) (HP ArcSight)" on page 941 for more information.
Severity	The incident Severity represents the seriousness calculated for the incident. Use the Severity attribute to specify the severity that should be assigned to the incident you are configuring. Possible values are described below: **Normal** - Indicates there are no known problems related to the associated object.

[1] Protocols used on Switches to configure multiple Interfaces (Aggregation Member Interfaces) to function as if they were one (an Aggregator Interface). When two Aggregator Interfaces establish a connection, that connection is an Aggregator Layer 2 Connection. The Aggregator Layer 2 Connection appears on Layer 2 Neighbor View maps as a thick line with an Interface icon at each end (representing the Aggregator Interface). When you double-click the thick line, it converts into multiple thin lines representing the participating Aggregation Member Layer 2 Connections with their Aggregation Member Interfaces at each end of the lines.

Interface Settings Enrichment Configuration Attributes , continued

Name	Description
	This Severity is meant to be informational. Generally, no action is needed for these incidents. **Warning** - Indicates there might be a problem related to the associated object. **Minor** - Indicates NNMi has detected problems related to the associated object that require further investigation. **Major** - Indicates NNMi has detected problems related to the associated object to be resolved before they become critical. **Critical** - Indicates NNMi has detected problems related to the associated object that require immediate attention.
Priority	Used to communicate the urgency of resolving the selected incident. You control this value. NNMi sets this value to null by default. The lower the number the higher the priority. Possible values are: 5 **None** 4 **Low** 3 **Medium** 2 **High** 1 **Top** **Note**: The icons are displayed only in table views.
Correlation Nature	Use the Correlation Nature to customize the Correlation Nature for this incident configuration. Possible values include: • Info • None • Root Cause • Secondary Root Cause • Symptom • Stream Correlation • Service Impact • Dedup Stream Correlation • Rate Stream Correlation See Incident Form: General Tab for more information.
Message	When configuring an incident, specify how the incident message appears in the

Interface Settings Enrichment Configuration Attributes , continued

Name	Description
Format	incident view. The string you specify in the Message Format attribute is visible in an incident view. **Note**: The incident message limit is 1024 characters. If you exceed this limit, NNMi truncates the value starting from the right. You can use any combination of default and custom attributes: "Valid Parameters for Configuring Incident Messages (Syslog Message) (HP ArcSight)" on page 946 "Include Custom Incident Attributes in Your Message Format (Syslog Message) (HP ArcSight)" on page 953
Assigned To	Use to specify the owner of any incident generated for this incident configuration. Click the Lookup icon and select Quick Find to select a valid user name. **Note**: You can also begin to type a valid user name and use the auto-complete feature to select the user name of interest.
Description	Use the Description attribute to provide additional information that you want to note about the current enhancement configuration. This description applies only to the enhancement configuration and does not appear when NNMi displays any associated incident. Type a maximum of 1024 characters. Alpha-numeric and special characters (~ ! @ # $ % ^ & * ( ) _+ -) are permitted.

Configure Custom Incident Attributes to Enrich an Incident Configuration (Interface Settings) (Syslog Message)(HP ArcSight)

The Custom Incident Attributes (CIAs) tab enables you to specify additional CIAs to be saved with an incoming incident. The persisted data might then be used as an argument to an action for the incident.

When creating a CIA for an incident configuration, you can specify any of the following values:

- Custom Attribute on the source node
- Custom Attribute on the interface (source object)

You also specify the Custom Incident Attribute name that will store this information.

For information about each Enrichment tab:

To create a Custom Incident Attribute to enrich an incident configuration:

1. Navigate to the **Syslog Message Configuration** form:
   a. From the workspace navigation panel, select the **Configuration** workspace.

   b. Expand the **Incidents** folder.

   c. Select **Syslog Message Configurations**.

   d. Do one of the following:

      i. To create an incident configuration, click the New icon, and continue.

      ii. To edit an incident configuration, select a row, click the Open icon, and continue.

      iii. To delete an incident configuration, select a row, and click the Delete icon.

2. Select **Interface Settings**.

3. Do one of the following:

   a. To create a new configuration, click the New icon.

   b. To edit an existing configuration, double-click the row representing the configuration you want to edit.

4. Make sure you configure the basic Interface Setting behavior. See "Configure Interface Settings for a Syslog Message Incident (HP ArcSight)" on page 954 for more information.

5. Select the **Enrichment** tab.

6. Do one of the following:

   a. To create an Enrichment configuration, click the New icon, and continue.

   b. To edit an Enrichment configuration,select a row, click the Open icon, and continue.

   c. To delete an Enrichment configuration, select a row and click the Delete icon.

7. Make sure you configure the Enrichment settings. See "Configure Incident Enrichment Settings for an Interface Group (Syslog Message)(HP ArcSight)" on page 963 for more information.

8. Navigate to the **Custom Incident Attributes** tab.

9. Do one of the following:

   a. To create a Custom Incident Attribute, click the New icon, and continue.

   b. To edit a Custom Incident Attribute, select a row, click the Open icon, and continue.

   c. To delete a Custom Incident Attribute, select a row and click the Delete icon.

10. Configure the Custom Incident Attribute (see the "Custom Incident Attribute" table).

11. Click **Save and Close** to save your changes and return to the previous form.

Custom Incident Attribute

Name	Description
Custom Incident Attribute Name	Name used to identify the Custom Incident Attribute you are configuring. The name limit is 255 characters. Alpha-numeric characters are permitted. No spaces or special characters (~ ! @ # $ % ^ & * ( ) _+ -) are permitted.

Custom Incident Attribute , continued

Name	Description
Type	Specifies whether you are using a Custom Attribute on a node or a Custom Attribute on an interface. Possible values are: • Node Custom Attribute • Interface Custom Attribute
Custom Attribute Name	Used to determine the value to be assigned to the Custom Incident Attribute you are configuring. Enter either of the following: • Name of the Custom Attribute on the source node • Name of the Custom Attribute on the interface (source object)

Configure a Payload Filter to Enrich an Incident Configuration (Interface Settings) (Syslog Message) (HP ArcSight)

The Payload Filter Editor enables you to create expressions that further refine the filters used to select the incidents to be enriched. Make sure to design any complex Payload Filters offline as a Boolean expression first. This method can help to minimize errors when entering your expressions using the Payload Filter editor.

To create a Payload Filter expression:

1. Navigate to the **Syslog Message Configuration** form:
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Expand the **Incidents** folder.
   c. Select **Syslog Message Configurations**.
   d. Do one of the following:
      i. To create an incident configuration, click the ✳ New icon, and continue.
      ii. To edit an incident configuration, select a row, click the 📂 Open icon, and continue.
      iii. To delete an incident configuration, select a row, and click the ✖ Delete icon.
2. Select the **Interface Settings** tab.
3. Do one of the following:
   a. To create a new configuration, click the ✳ New icon.
   b. To edit an existing configuration, double-click the row representing the configuration you want to edit.
4. Make sure you configure the basic Interface Setting behavior. See "Configure Interface Settings for a Syslog Message Incident (HP ArcSight)" on page 954 for more information.
5. Select the **Enrichment** tab.
6. Do one of the following:

   a. To create an Enrichment configuration, click the ✳ New icon, and continue.

   b. To edit an Enrichment configuration, select a row, click the Open icon, and continue.

   c. To delete an Enrichment configuration, select a row and click the ✖ Delete icon.

7. Make sure you configure the Enrichment settings. See "Configure Incident Enrichment Settings for an Interface Group (Syslog Message)(HP ArcSight)" on page 963 for more information.

8. Select the **Payload Filter** tab.

9. Define your Payload Filter (see table). Also see "Guidelines for Creating a Payload Filter".

   a. Plan out the logic needed for your Filter String.

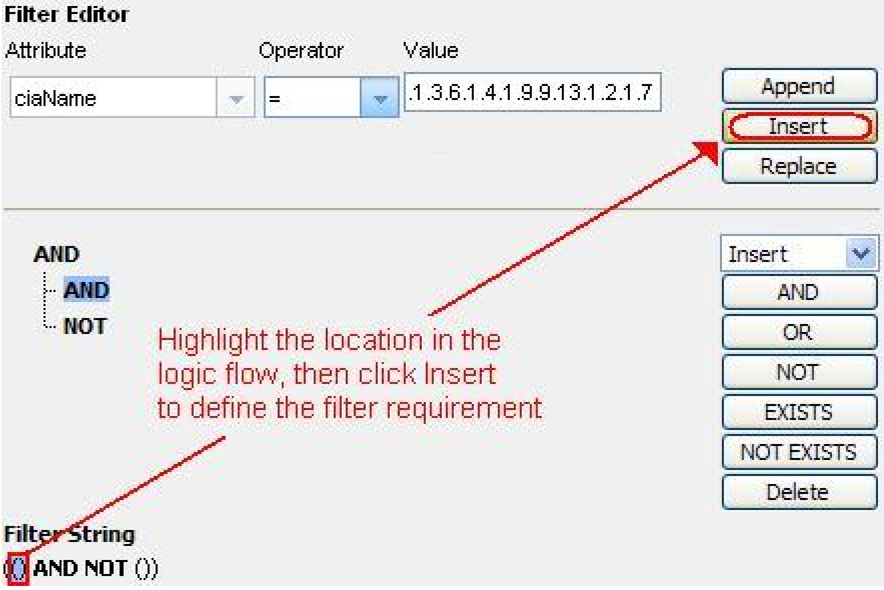
   b. Use the buttons on the bottom half of the Additional Filters Editor to establish the logic structure.

   For example, to establish the following structure, select **Insert**, then click **AND**, then **NOT**, and then **AND** a second time:

   ```
   (( ) AND NOT ( ))
   ```

   c. Now place your cursor in a location within the displayed Filter String, and use the top half of the filter editor to define the parameters of the highlighted filter requirement.

   For example, select a set of parentheses and use the Insert button to specify the filter requirement within those parentheses:



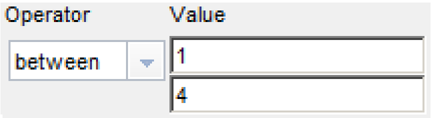
10. Click **Save and Close**.

11. Click **Save and Close** to save your changes and return to the previous form.

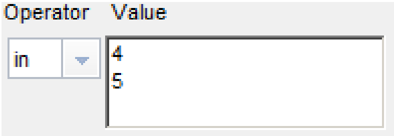
Payload Filter Editor Components

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the

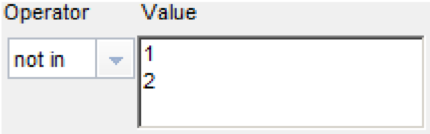
Payload Filter Editor Components, continued

Attribute	Description
	following: • ciaName • ciaValue
Operator	Valid operators are described below. • **=** Finds all values equal to the value specified. Click here for an example. Example: `ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7` matches any incident that contains a varbind with the name value **.1.3.6.1.4.1.9.9.13.1.2.1.7**. • **!=** Finds all values not equal to the value specified. Click here for an example. Example: `ciaName != .1.3.6.1.4.1.9.9.13.1.2.1.7` matches any incident that contains a varbind with the name value other than **1.3.6.1.4.1.9.9.13.1.2.1.7**. • **<** Finds all values less than the value specified. Click here for an example. Example: `ciaValue < 6` matches any incident with a varbind value less than **6**. • **<=** Finds all values less than or equal to the value specified. Click here for an example. Example: `ciaValue <= 6` matches any incident with a varbind value less than or equal to **6**. • **>** Finds all values greater than the value specified. Click here for an example. Example: `ciaValue > 4` matches any incident with a varbind value greater than **4**. • **>=** Finds all values greater than or equal to the value specified. Click here for an example. Example: `ciaValue >= 4` matches any incident that contains a varbind with values greater than or equal to **4**. • **between** Finds all traps or events that include a varbind with a value equal to and between the two values specified. Click here for an example. Example: `ciaValue between`  Operator: between Value: 1 4 matches any incident that contains a varbind value equal to or greater than **1** and equal to or less than **4**. **Note**: As shown in the example, each value must be entered on a separate line. • **in** Finds any match to at least one value in a list of values. Click here for an example.

Payload Filter Editor Components, continued

Attribute	Description
	Example: `ciaValue in`  Operator Value in 4 5 matches any incident with a varbind value of either **4** or **5**. **Note**: As shown in the example, each value must be entered on a separate line. NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example (**4, 5**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line. • **is not null** Finds all non-blank values. Click here for an example. Example: `ciaValue is not null` matches any incident with a varbind that contains a value. • **is null** Finds all blank values. Click here for an example. Example: `ciaValue is null` matches any incident with a varbind that does not have a value. • **like** Finds matches using wildcard characters. Click here for more information about using wildcard characters. The period asterisk (.*) characters mean *any number of characters of any type at this location*. The period (.) character means *any single character of any type at this location*. **Note**: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below. Examples: `ciaName like  \Q.1.3.6.1.4.1.9.9\E.*` finds all traps or events that contain varbind names that begin with **.1.3.6.1.4.1.9.9** and (optionally) end with any number of characters. `ciaValue like .*Chicago.*` finds all traps or events that contain a varbind value that includes the string **Chicago**. • **not between** Finds all values except those between the two values specified. Click here for an example. Example: `ciaValue not between 5 8` matches an incident that contains a varbind with the values less than **5** or greater than **8**. • **not in** Finds all values except those included in the list of values. Click here for an example. Example:

Payload Filter Editor Components, continued

Attribute	Description
	`ciaValue not in`  Operator: not in Value: 1 2 matches any incident that contains a varbind with values other than **1** and **2**. **Note**: As shown in the example, each value must be entered on a separate line. NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example, (**1, 2**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line. • **not like** Finds all that do not have the values specified (using wildcard strings). Click here for an example. The period asterisk (.*) characters mean *any number of characters of any type at this location*. The period (.) character means *any single character of any type at this location*. **Note**: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below. Example: `ciaName not like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that does not begin with **.1.3.6.1.4.1.9.9**. `ciaValue not like .*Chicago.*` finds all traps or events that do not contain a varbind value that includes the string **Chicago**.
Value	The value for which you want NNMi to search. Note the following: • The values you enter are case sensitive. • NNMi displays a variable number of value fields depending on the Operator selected. For example, the `between` Operator causes two value fields to be displayed. • The `between, in` and `not in` operators require that each value be entered on a separate line.

Additional Filters Editor Buttons

Button	Description
Append	Appends the current expression (Attribute, Operator,and Value) to the selected expression already included in the Filter String.
Insert	Inserts the current expression (Attribute, Operator,and Value) in front of the

Additional Filters Editor Buttons, continued

Button	Description
	cursor location within the Filter String.
Replace	Replaces the selected expression with the expression displayed in the Attribute, Operator, and Value fields.
AND	Appends, inserts, or replaces the AND Boolean Operator in the selected cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
OR	Appends, Inserts, or replaces the OR Boolean Operator in the current cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
NOT	Can be used in any part of the Filter String to specify that NNMi should exclude interfaces with values that pass the expression that immediately follows the `NOT`. For example, when evaluating the following expression, NNMi includes interfaces with (interface description) `ifDesc` containing **VLAN**, and excludes any Interfaces that have **VLAN10** for the (interface name) `ifName` value: `(ifDesc like VLAN AND NOT (ifName=VLAN10))` **Note**: View the expression displayed under Filter String to see the logic of the expression as it is created .
EXISTS	Used for filters that include Capabilities or Custom Attribute names and values in the Filer String. Indicates that you want NNMi to consider interfaces that do not have any Capabilities or Custom Attributes when evaluating the Filter String. **Note**: If you include Capabilities or Custom Attribute names and values in the Filer String, but do not use **EXISTS** or **NOT EXISTS**, NNMi excludes from its search interfaces that do not include Capabilities or Custom Attributes. For example, when evaluating the following Filter String, NNMi includes interfaces with (interface description) `ifDesc` containing **VLAN**, as well as any Interfaces Custom Attribute Role value is **LAN Connection to Oracle Server**: `(ifDesc like VLAN OR EXISTS((customAttrName=Role AND customAttrValue=LAN Connection to Oracle Server)))` **Note**: View the expression displayed under Filter String to see the logic of the expression as it is created.
NOT EXISTS	Used for filters that include Capabilities or Custom Attribute names and values in the Filer String. Indicates that you want NNMi to consider interfaces that do not have any Capabilities or Custom Attributes when evaluating the Filter String, but

Additional Filters Editor Buttons, continued

Button	Description
	exclude the interfaces that match the expression that follows the **NOT EXISTS**. Note: If you include Capabilities or Custom Attribute names and values in the Filer String, but do not use **EXISTS** or **NOT EXISTS**, NNMi excludes from its search interfaces that do not include Capabilities or Custom Attributes. For example, when evaluating the following expression, NNMi includes interfaces with (interface description) `ifDesc` containing **VLAN**, and excludes any Interfaces that have the Custom Attribute Role and that Role value is **LAN Connection to Oracle Server**: `(ifDesc like VLAN OR NOT EXISTS((customAttrName=Role AND customAttrValue=LAN Connection to Oracle Server)))` **Note**: View the expression displayed under Filter String to see the logic of the expression as it is created.
Delete	Deletes the selected expression. **Note**: If the Boolean Operator is selected, the Additional Filters Editor deletes all expressions associated with the Boolean Operator.

Configure Incident Dampening Settings for an Interface Group (Syslog Message) (HP ArcSight)

Note: Interface Settings override any other Dampening settings for this incident, including those from the Node Settings tab.

NNMi enables you to delay the following for an incident configuration based on the Source Object's participation in an Interface Group:

- Execution of Incident Actions
- Execution of Diagnostics (*HP Network Node Manager iSPI Network Engineering Toolset Software \ NNM iSPI NET*)
- Appearance within Incident views in the NNMi Console

Note: You can also configure the Dampening settings based on the Source Node's participation in a Node Group. See "Configure Incident Dampening Settings for a Node Group (Syslog Message) (HP ArcSight)" on page 1012 for more information.

Tip: See "Create Interface Groups" on page 321 for more information about Interface Groups.

For information about each Interface Settings tab:

When using the Dampening configuration, note the following:

- NNMi initially assigns incidents with Dampening settings configured a Lifecycle State of DAMPENED.
- After the Dampen Interval has passed, NNMi changes the Lifecycle State to REGISTERED.

See About the Incident Lifecycle for more information about Lifecycle State.

To configure the Dampening settings based on an Interface Group:

1. Navigate to the **Syslog Message Configuration** form:
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Expand the **Incidents** folder.
   c. Select **Syslog Message Configurations**.
   d. Do one of the following:
      i. To create an incident configuration, click the ✳ New icon, and continue.
      ii. To edit an incident configuration, select a row, click the 📂 Open icon, and continue.
      iii. To delete an incident configuration, select a row, and click the ✖ Delete icon.
2. Select the **Interface Settings** tab.
3. Do one of the following:
   a. To create a new configuration, click the ✳ New icon.
   b. To edit an existing configuration, select a row, click the 📂 Open icon, and continue.
4. Make sure you configure the basic Interface Setting behavior. See "Configure Interface Settings for a Syslog Message Incident (HP ArcSight)" on page 954 for more information.
5. Select the **Dampening** tab.
6. Configure the desired Dampening behavior (see table).
7. Click 💾 **Save and Close** to save your changes and return to the previous form.

Interface Settings Dampening Configuration Attributes

Name	Description
Enable	Use this attribute to temporarily disable an incident's dampening settings: **Enable** □ = Temporarily disable the selected configuration. **Enable** ☑ = Enable the selected configuration.
Hour	Specifies the number of hours to be used for the dampen interval.
Minutes	Specifies the number of minutes to be used for the dampen interval. **Note:** The maximum dampen interval is 60 minutes.
Seconds	Specifies the number of seconds to be used for the dampen interval.
Payload Filter	The Payload Filter Editor enables you to create expressions that further refine the filters used to select the incidents to be suppressed, enriched, or dampened. Make sure to design any complex Payload Filters offline as a Boolean expression first. This method can help to minimize errors when entering your expressions using the Payload Filter

Interface Settings Dampening Configuration Attributes , continued

Name	Description

	editor.
	When creating a Payload Filter, note the following:
	• Payload Filter expressions for the `like` and `not like` operators use the syntax defined for java regular expressions (java.util.regex Pattern class).
	• You must use a `ciaName` that already exists in the trap or event you are configuring.
	• Each set of expressions associated with a Boolean Operator is treated as if it were enclosed in parentheses and evaluated together.
	• View the expression displayed under **Filter String** to see the logic of the expression as it is created.
	• The `AND` and `OR` Boolean Operators must contain at least two expressions as shown in the example below.
	The following example filters incidents on voltage state:
	`AND`
	`    ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7`
	`    ciaValue = 5`
	NNMi evaluates the expression above as follows:
	`(ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7 AND ciaValue = 5)`
	NNMi finds all incidents with a varbind `.1.3.6.1.4.1.9.9.13.1.2.1.7` value of **5**.
	• The placement of your cursor and the subsequent text that is selected is important when performing operations using the Payload Filter Editor. For example, you append to, replace, or change the indentation of the expression that is selected.
	• The placement of your cursor and the subsequent text that is selected is especially important when adding your Boolean operators.
	• You can include more than one varbind in the same Payload Filter expression as shown in the following example:
	`((ciaName like \Q.1.3.6.1.4.1.9.9\E.* AND ciaValue = 25) AND (ciaName like \Q.1.3.6.1.2.1.2.2.1.1.3\E.* AND ciaValue = 3))`
	In this example, a given trap must meet each of the following criteria:
	▪ Contain a varbind whose Object Identifier (OID) matches the regular expression `\Q.1.3.6.1.4.1.9.9\E.*` and has a value of `25`.
	▪ Contain a varbind whose OID matches the regular expression `\Q.1.3.6.1.2.1.2.2.1.1.3\E.*` and has a value of `3`.

Interface Settings Dampening Configuration Attributes , continued

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td></td><td>

Payload Filter Editor Components

<table>
<tr><th>Attribute</th><th>Description</th></tr>
<tr><td>Attribute</td><td>The attribute name on which NNMi searches. Filterable attributes include the following:
• ciaName
• ciaValue</td></tr>
<tr><td>Operator</td><td>Valid operators are described below.
• **=** Finds all values equal to the value specified. Click here for an example.
Example: `ciaName=.1.3.6.1.4.1.9.9.13.1.2.1.7` matches any incident that contains a varbind with the name value of **.1.3.6.1.4.1.9.9.13.1.2.1.7**.
• **!=** Finds all values not equal to the value specified. Click here for an example.
Example: `ciaName! = .1.3.6.1.4.1.9.9.13.1.2.1.7` matches any incident that contains a varbind with a name value other than **1.3.6.1.4.1.9.9.13.1.2.1.7**.
• **<** Finds all values less than the value specified. Click here for an example.
Example: `ciaValue < 6` matches any incident that contains a varbind with a value less than **6**.
• **<=** Finds all values less than or equal to the value specified. Click here for an example.
Example: `ciaValue <= 6` matches any incident that contains a varbind with a value less than or equal to **6**.
• **>** Finds all values greater than the value specified. Click here for an example.
Example: `ciaValue > 4` matches any incident that contains a varbind with a value greater than **4**.
• **>=** Finds all values greater than or equal to the value specified. Click here for an example.
Example: `ciaValue >= 4` matches any incident that contains a varbind with values greater than or equal to **4**.
• **between** Finds all values equal to and between the two values specified. Click here for an example.
Example: `ciaValue between`</td></tr>
</table>

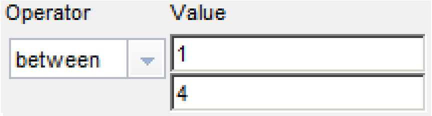
</td></tr>
</table>

Interface Settings Dampening Configuration Attributes , continued

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td></td><td>

Payload Filter Editor Components, continued

<table>
<tr><th>Attribute</th><th>Description</th></tr>
<tr><td></td><td>



Operator: between
Value: 1
4

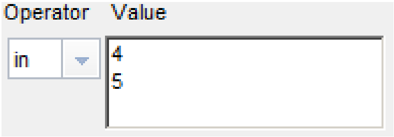
matches any incident that contains a varbind value equal to or greater than **1** and equal to or less than **4**.

Note: As shown in the example, each value must be entered on a separate line.

- **in** Finds any match to at least one value in a list of values. Click here for an example.

  Example:

  `ciaValue in`

  Operator: in
  Value: 4
  5

  matches any incident with a varbind value of either **4** or **5**.

  Note: As shown in the example, each value must be entered on a separate line.

  NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example (**4, 5**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line.

- **is not null** Finds all non-blank values. Click here for an example.

  Example: `ciaValue is not null` matches any incident with a varbind that contains a value.

- **is null** Finds all blank values. Click here for an example.

  Example: `ciaValue is null` matches any incident with a varbind that does not contain a value.

- **like** Finds matches using the syntax defined for java regular expressions. See the Pattern (Java Platform SE6) API documentation at : `http://download.oracle.com/javase/6/docs/api/java/util/regex/Pattern.html` for more information. Click here for more information.

  The period asterisk (.*) characters mean *any number of characters of any*

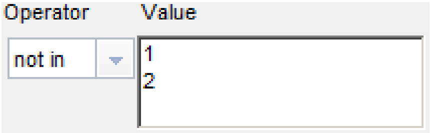
</td></tr>
</table>

</td></tr>
</table>

Interface Settings Dampening Configuration Attributes , continued

Name	Description

Payload Filter Editor Components, continued

Attribute	Description
	type at this location. The period (.) character means *any single character of any type at this location*. **Note**: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below. Example: `ciaName like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that begins with 1.3.6.1.4.1.9.9 and (optionally) ends with any number of characters. `ciaValue like .*Chicago.*` finds all traps or events that contain a varbind value that includes the string **Chicago**. • **not between** Finds all values except those between the two values specified. Click here for an example. Example: `ciaValue not between 5 8` matches an incident that contains a varbind with the values less than **5** or greater than **8** . • **not in** Finds all values except those included in the list of values. Click here for an example. Example: `ciaValue not in`  Operator: not in Value: 1 2 matches any incident that contains a varbind with values other than **1** and **2**. **Note**: As shown in the example, each value must be entered on a separate line. NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example, (**1, 2**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line. • **not like** Finds all that do not have the values specified using the syntax defined for Java regular expressions. See the Pattern (Java Platform SE6)

Interface Settings Dampening Configuration Attributes , continued

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td></td><td>

Payload Filter Editor Components, continued

<table>
<tr><th>Attribute</th><th>Description</th></tr>
<tr><td></td><td>
API documentation at: <code>http://download.oracle.com/javase/6/docs/api/java/util/regex/Pattern.html</code> for more information. Click here for an example.
The period asterisk (.*) characters mean <i>any number of characters of any type at this location</i>.
The period (.) character means <i>any single character of any type at this location</i>.
<b>Note</b>: To include literal string values in the Value attribute, enclose the string value in \Q&lt;<i>literal_value</i>&gt;\E as shown in the Examples listed below.
Example:
<code>ciaName not like \Q.1.3.6.1.4.1.9.9\E.*</code> matches any incident that contains a varbind name value that does not begin with <b>.1.3.6.1.4.1.9.9</b> and (optionally) ends with any number of characters.
<code>ciaValue not like .*Chicago.*</code> finds all traps or events that do not contain a varbind value that includes the string <b>Chicago</b>.
</td></tr>
<tr><td>Value</td><td>
The value for which you want NNMi to search.
Note the following:
• The values you enter are case sensitive.
• NNMi displays a variable number of value fields depending on the Operator selected. For example, the <code>between</code> Operator causes two value fields to be displayed.
• The <code>between, in</code> and <code>not in</code> operators require that each value be entered on a separate line.
</td></tr>
</table>

Payload Filter Editor Buttons

<table>
<tr><th>Button</th><th>Description</th></tr>
<tr><td>Append</td><td>Appends the current expression (Attribute, Operator,and Value) to the selected expression already included in the filter string.</td></tr>
<tr><td>Insert</td><td>Inserts the current expression (Attribute, Operator,and Value) in front of the cursor location within the Filter String.</td></tr>
<tr><td>Replace</td><td>Replaces the selected expression with the expression displayed in</td></tr>
</table>

</td></tr>
</table>

Interface Settings Dampening Configuration Attributes , continued

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td></td><td>

Payload Filter Editor Buttons, continued

<table>
<tr><th>Button</th><th>Description</th></tr>
<tr><td></td><td>the Attribute, Operator, and Value fields.</td></tr>
<tr><td>AND</td><td>Inserts the AND Boolean Operator in the selected cursor location.
<b>Note</b>: View the expression displayed under <b>Filter String</b> to see the logic of the expression as it is created.</td></tr>
<tr><td>OR</td><td>Inserts the OR Boolean Operator in the current cursor location.
<b>Note</b>: View the expression displayed under <b>Filter String</b> to see the logic of the expression as it is created.</td></tr>
<tr><td>NOT</td><td>Can be used in any part of the Filter String to specify that NNMi should exclude interfaces with values that pass the expression that immediately follows the <code>NOT</code>.
For example, when evaluating the following expression, NNMi includes interfaces with (interface description) <code>ifDesc</code> containing <b>VLAN</b>, and excludes any Interfaces that have <b>VLAN10</b> for the (interface name) <code>ifName</code> value:
<code>(ifDesc like VLAN AND NOT (ifName=VLAN10))</code>
<b>Note</b>: View the expression displayed under Filter String to see the logic of the expression as it is created .</td></tr>
<tr><td>EXISTS</td><td>Used for filters that include Capabilities or Custom Attribute names and values in the Filer String.
Indicates that you want NNMi to consider interfaces that do not have any Capabilities or Custom Attributes when evaluating the Filter String.
<b>Note</b>: If you include Capabilities or Custom Attribute names and values in the Filer String, but do not use <b>EXISTS</b> or <b>NOT EXISTS</b>, NNMi excludes from its search interfaces that do not include Capabilities or Custom Attributes.
For example, when evaluating the following Filter String, NNMi includes interfaces with (interface description) <code>ifDesc</code> containing <b>VLAN</b>, as well as any Interfaces Custom Attribute Role value is <b>LAN Connection to Oracle Server</b>:
<code>(ifDesc like VLAN OR EXISTS((customAttrName=Role AND customAttrValue=LAN Connection to Oracle Server)))</code>
<b>Note</b>: View the expression displayed under Filter String to see the</td></tr>
</table>

</td></tr>
</table>

Interface Settings Dampening Configuration Attributes , continued

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td></td><td>

Payload Filter Editor Buttons, continued

<table>
<tr><th>Button</th><th>Description</th></tr>
<tr><td></td><td>logic of the expression as it is created.</td></tr>
<tr><td>NOT EXISTS</td><td>Used for filters that include Capabilities or Custom Attribute names and values in the Filer String. Indicates that you want NNMi to consider interfaces that do not have any Capabilities or Custom Attributes when evaluating the Filter String, but exclude the interfaces that match the expression that follows the <b>NOT EXISTS</b>.
<b>Note</b>: If you include Capabilities or Custom Attribute names and values in the Filer String, but do not use <b>EXISTS</b> or <b>NOT EXISTS</b>, NNMi excludes from its search interfaces that do not include Capabilities or Custom Attributes.
For example, when evaluating the following expression, NNMi includes interfaces with (interface description) <code>ifDesc</code> containing <b>VLAN</b>, and excludes any Interfaces that have the Custom Attribute Role and that Role value is <b>LAN Connection to Oracle Server</b>:
<code>(ifDesc like VLAN OR NOT EXISTS ((customAttrName=Role AND customAttrValue=LAN Connection to Oracle Server)))</code>
<b>Note</b>: View the expression displayed under Filter String to see the logic of the expression as it is created.</td></tr>
<tr><td>Delete</td><td>Deletes the selected expression.
<b>Note</b>: If the Boolean Operator is selected, the Payload Filter Editor deletes all expressions associated with the Boolean Operator.</td></tr>
</table>

</td></tr>
</table>

Configure Incident Actions for an Interface Group (Syslog Message) (HP ArcSight)

Note: Interface Settings override any other Actions settings for this incident, including those from the Node Settings tab.

For information about each Interface Settings tab:

NNMi enables you to configure incident actions based on a Source Object's participation in an Interface Group.

You can configure actions to automatically run at any point in the incident lifecycle. For example, you might want to configure an action to occur when an incident of the type you are configuring is generated (**Registered**). When an incident is generated, you might want to automatically open a trouble ticket or send email or page your network operator. After the incident is **Closed**, you might want to automatically close the trouble ticket.

> **Note:** Your actions will not be executed until you enable the Actions configuration by either clicking Enable ☑ on the Actions tab or using the **Actions** → **Enable Configuration** option.

You can configure actions for incidents generated from SNMP traps, Syslog Messages (HP ArcSightonly), NNM 6.x or 7.x events, and the NNMi Management Events. Any time an incident configuration changes, the action directory is rescanned and any Jython files are reloaded to the NNMi database. See "Configure Actions for a Syslog Message Incident (HP ArcSight)" on page 1063 for more information about the actions directory.

Tip: Copy any required Jython files to the NNMi actions directory before you configure an incident action. New or updated actions are loaded into NNMi only when an incident configuration is updated or created. See "Lifecycle Transition Action Form (Syslog Message) (HP ArcSight)" on page 1064 for the location of the NNMi action directory.

When the action runs, output is logged to the `incidentActions.*.*.log` file. See "Verify that NNMi Services are Running" on page 85 for more information about log files and where they are located.

To configure an automatic action for an incident:

1. Navigate to the **Syslog Message Configuration** form.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Expand the **Incidents** folder.
   c. Select **Syslog Message Configurations**.
   d. Do one of the following:
      i. To create a new incident configuration, click the ✱ New icon.
      ii. To edit an existing incident configuration, select a row, click the 📂 Open icon, and continue.
2. Select the **Interface Settings** tab.
3. Do one of the following:
   a. To create a new configuration, click the ✱ New icon.
   b. To edit an existing configuration, select a row, click the 📂 Open icon, and continue.
4. Make sure you configure the basic Interface Setting behavior. See "Configure Interface Settings for a Syslog Message Incident (HP ArcSight)" on page 954for more information.
5. Select the **Actions** tab.
6. From the **Lifecycle Actions** table toolbar, do one of the following:
   - To create an Action configuration, click the ✱ New icon, and continue.
   - To edit an Action configuration, select a row, click the 📂 Open icon, and continue.
   - To delete an Action configuration, select a row, and click the ✖ Delete icon.
7. In the "Lifecycle Transition Action Form (Syslog Message) (HP ArcSight)" on page 1064, provide the required information.

8. Click **Save and Close** to save your changes and return to the previous form.

   The next time the lifecycle changes, NNMi launches the action associated with the lifecycle for the incident of that type.

Configure a Payload Filter for an Incident Action (Interface Settings) (Syslog Message) (HP ArcSight)

The Payload Filter Editor enables you to create expressions that further refine the filters used to select the incidents that cause the configured action to run. Make sure to design any complex Payload Filters offline as a Boolean expression first. This method can help to minimize errors when entering your expressions using the Payload Filter editor.

To create a Payload Filter expression:

1. Navigate to the **Syslog Message Configuration** form:
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Expand the **Incidents** folder.
   c. Select **Syslog Message Configurations**.
   d. Do one of the following:
      i. To create an incident configuration, click the ✱ New icon, and continue.
      ii. To edit an incident configuration, select a row, click the Open icon, and continue.
      iii. To delete an incident configuration, select a row, and click the ✖ Delete icon.
2. Select the **Interface Settings** tab.
3. Do one of the following:
   a. To create a new configuration, click the ✱ New icon.
   b. To edit an existing configuration, double-click the row representing the configuration you want to edit.
4. Make sure you configure the basic Interface Setting behavior. See "Configure Interface Settings for a Syslog Message Incident (HP ArcSight)" on page 954 for more information.
5. Select the **Actions** tab.
6. Do one of the following:
   a. To create an Action configuration, click the ✱ New icon, and continue.
   b. To edit an Action configuration, double-click the row representing the configuration you want to edit, and continue.
   c. To delete an Action configuration, select a row, and click the ✖ Delete icon.
7. Make sure the Action settings are configured. See "Configure Incident Actions for an Interface Group (Syslog Message) (HP ArcSight)" on page 983 for more information.

8. Select the **Payload Filter** tab.

9. Define your Payload Filter (see table). Also see "Guidelines for Creating a Payload Filter".

   a. Plan out the logic needed for your Filter String.

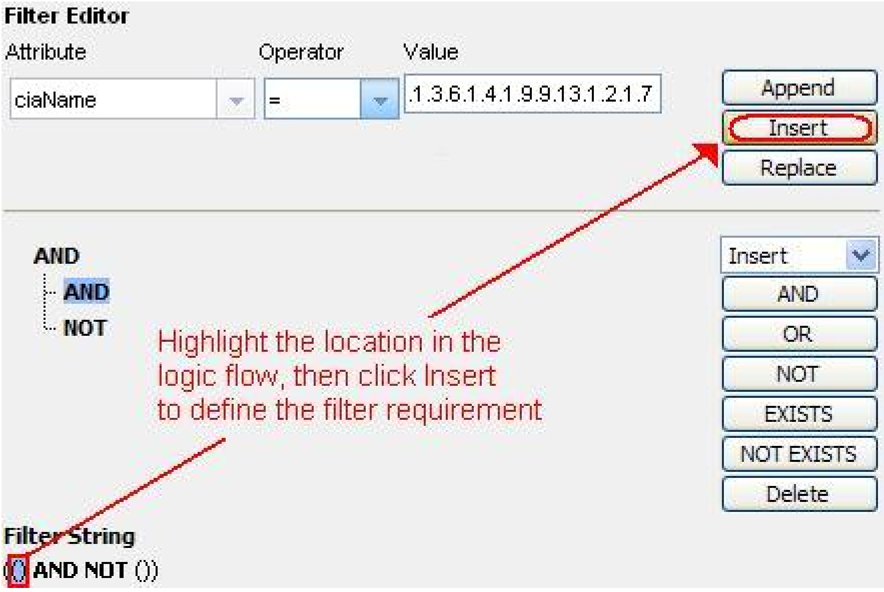
   b. Use the buttons on the bottom half of the Additional Filters Editor to establish the logic structure.

      For example, to establish the following structure, select **Insert**, then click **AND**, then **NOT**, and then **AND** a second time:

      ```
      (( ) AND NOT ( ))
      ```

   c. Now place your cursor in a location within the displayed Filter String, and use the top half of the filter editor to define the parameters of the highlighted filter requirement.

      For example, select a set of parentheses and use the Insert button to specify the filter requirement within those parentheses:



Filter Editor

Attribute: ciaName

Operator: =

Value: .1.3.6.1.4.1.9.9.13.1.2.1.7

Highlight the location in the logic flow, then click Insert to define the filter requirement

Filter String
(() AND NOT ())

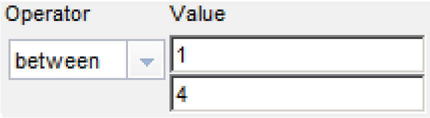
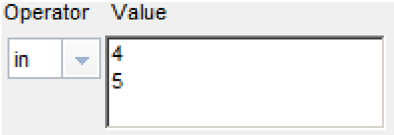
10. Click **Save and Close**.

11. Click **Save and Close** to save your changes and return to the previous form.

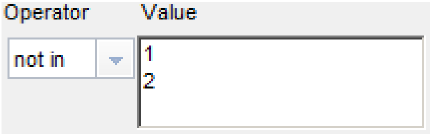
Payload Filter Editor Components

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: • ciaName • ciaValue
Operator	Valid operators are described below. • **=** Finds all values equal to the value specified. Click here for an example.

Payload Filter Editor Components, continued

Attribute	Description
	Example: `ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7` matches any incident that contains a varbind with the name value **.1.3.6.1.4.1.9.9.13.1.2.1.7**. • **!=** Finds all values not equal to the value specified. Click here for an example. Example: `ciaName != .1.3.6.1.4.1.9.9.13.1.2.1.7` matches any incident that contains a varbind with the name value other than **1.3.6.1.4.1.9.9.13.1.2.1.7**. • **<** Finds all values less than the value specified. Click here for an example. Example: `ciaValue < 6` matches any incident with a varbind value less than **6**. • **<=** Finds all values less than or equal to the value specified. Click here for an example. Example: `ciaValue <= 6` matches any incident with a varbind value less than or equal to **6**. • **>** Finds all values greater than the value specified. Click here for an example. Example: `ciaValue > 4` matches any incident with a varbind value greater than **4**. • **>=** Finds all values greater than or equal to the value specified. Click here for an example. Example: `ciaValue >= 4` matches any incident that contains a varbind with values greater than or equal to **4**. • **between** Finds all traps or events that include a varbind with a value equal to and between the two values specified. Click here for an example. Example: `ciaValue between`  Operator: between Value: 1 4 matches any incident that contains a varbind value equal to or greater than **1** and equal to or less than **4**. **Note**: As shown in the example, each value must be entered on a separate line. • **in** Finds any match to at least one value in a list of values. Click here for an example. Example: `ciaValue in`  Operator: in Value: 4 5 matches any incident with a varbind value of either **4** or **5**.

Payload Filter Editor Components, continued

Attribute	Description
	Note: As shown in the example, each value must be entered on a separate line. NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example (**4, 5**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line. - **is not null** Finds all non-blank values. Click here for an example. Example: `ciaValue is not null` matches any incident with a varbind that contains a value. - **is null** Finds all blank values. Click here for an example. Example: `ciaValue is null` matches any incident with a varbind that does not have a value. - **like** Finds matches using wildcard characters. Click here for more information about using wildcard characters. The period asterisk (.*) characters mean *any number of characters of any type at this location*. The period (.) character means *any single character of any type at this location*. **Note**: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below. Examples: `ciaName like  \Q.1.3.6.1.4.1.9.9\E.*` finds all traps or events that contain varbind names that begin with **.1.3.6.1.4.1.9.9** and (optionally) end with any number of characters. `ciaValue like .*Chicago.*` finds all traps or events that contain a varbind value that includes the string **Chicago**. - **not between** Finds all values except those between the two values specified. Click here for an example. Example: `ciaValue not between 5 8` matches an incident that contains a varbind with the values less than **5** or greater than **8**. - **not in** Finds all values except those included in the list of values. Click here for an example. Example: `ciaValue not in`  Operator not in Value 1 2 matches any incident that contains a varbind with values other than **1** and **2**. **Note**: As shown in the example, each value must be entered on a separate line.

Payload Filter Editor Components, continued

Attribute	Description
	NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example, (**1, 2**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line. • **not like** Finds all that do not have the values specified (using wildcard strings). Click here for an example. The period asterisk (.*) characters mean *any number of characters of any type at this location*. The period (.) character means *any single character of any type at this location*. **Note**: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below. Example: `ciaName not like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that does not begin with **.1.3.6.1.4.1.9.9**. `ciaValue not like .*Chicago.*` finds all traps or events that do not contain a varbind value that includes the string **Chicago**.
Value	The value for which you want NNMi to search. Note the following: • The values you enter are case sensitive. • NNMi displays a variable number of value fields depending on the Operator selected. For example, the `between` Operator causes two value fields to be displayed. • The `between, in` and `not in` operators require that each value be entered on a separate line.

Additional Filters Editor Buttons

Button	Description
Append	Appends the current expression (Attribute, Operator,and Value) to the selected expression already included in the Filter String.
Insert	Inserts the current expression (Attribute, Operator,and Value) in front of the cursor location within the Filter String.
Replace	Replaces the selected expression with the expression displayed in the Attribute, Operator, and Value fields.
AND	Appends, inserts, or replaces the AND Boolean Operator in the selected cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.

Additional Filters Editor Buttons, continued

Button	Description
OR	Appends, Inserts, or replaces the OR Boolean Operator in the current cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
NOT	Can be used in any part of the Filter String to specify that NNMi should exclude interfaces with values that pass the expression that immediately follows the `NOT`. For example, when evaluating the following expression, NNMi includes interfaces with (interface description) `ifDesc` containing **VLAN**, and excludes any Interfaces that have **VLAN10** for the (interface name) `ifName` value: `(ifDesc like VLAN AND NOT (ifName=VLAN10))` **Note**: View the expression displayed under Filter String to see the logic of the expression as it is created .
EXISTS	Used for filters that include Capabilities or Custom Attribute names and values in the Filer String. Indicates that you want NNMi to consider interfaces that do not have any Capabilities or Custom Attributes when evaluating the Filter String. **Note**: If you include Capabilities or Custom Attribute names and values in the Filer String, but do not use **EXISTS** or **NOT EXISTS**, NNMi excludes from its search interfaces that do not include Capabilities or Custom Attributes. For example, when evaluating the following Filter String, NNMi includes interfaces with (interface description) `ifDesc` containing **VLAN**, as well as any Interfaces Custom Attribute Role value is **LAN Connection to Oracle Server**: `(ifDesc like VLAN OR EXISTS((customAttrName=Role AND customAttrValue=LAN Connection to Oracle Server)))` **Note**: View the expression displayed under Filter String to see the logic of the expression as it is created.
NOT EXISTS	Used for filters that include Capabilities or Custom Attribute names and values in the Filer String. Indicates that you want NNMi to consider interfaces that do not have any Capabilities or Custom Attributes when evaluating the Filter String, but exclude the interfaces that match the expression that follows the **NOT EXISTS**. Note: If you include Capabilities or Custom Attribute names and values in the Filer String, but do not use **EXISTS** or **NOT EXISTS**, NNMi excludes from its search interfaces that do not include Capabilities or Custom Attributes. For example, when evaluating the following expression, NNMi includes interfaces with (interface description) `ifDesc` containing **VLAN**, and excludes any Interfaces that have the Custom Attribute Role and that Role value is **LAN Connection to Oracle Server**:

Additional Filters Editor Buttons, continued

Button	Description
	`(ifDesc like VLAN OR NOT EXISTS((customAttrName=Role AND customAttrValue=LAN Connection to Oracle Server)))` **Note**: View the expression displayed under Filter String to see the logic of the expression as it is created.
Delete	Deletes the selected expression. **Note**: If the Boolean Operator is selected, the Additional Filters Editor deletes all expressions associated with the Boolean Operator.

Configure Node Settings for a Syslog Message Incident (HP ArcSight)

Note: Node Settings override any other Suppression, Enrichment, Dampen, Action, or Diagnostics Selections configuration settings, except those configured on the Interface Settings tab.

NNMi enables you to apply an incident configuration to a Source Node based on the Source Node's participation in a Node Group. If the Source Node is not a member of the Node Group specified, the incident is neither displayed nor stored in the NNMi database.

Tip: See "Create Node Groups" on page 295 for more information about Node Groups.

For information about each Node Settings tab:

For information about each Syslog Message tab:

To apply an incident configuration to a Source Node based on the Source Node's Node Group:

1. Navigate to the **Syslog Message Configuration** form:
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Expand the **Incidents** folder.
   c. Select **Syslog Message Configurations**.
   d. Do one of the following:
      i. To create an incident configuration, click the ✳ New icon, and continue.
      ii. To edit an incident configuration, select a row, click the 📂 Open icon, and continue.
      iii. To delete an incident configuration, select a row, and click the ✖ Delete icon.
2. Select the **Node Settings** tab.
3. Do one of the following:
   a. To create a new configuration, click the ✳ New icon.
   b. To edit an existing configuration, double-click the row representing the configuration you want to edit.

4. Configure the desired Node Settings (see table).

5. Click **Save and Close** to save your changes and return to the previous form.

Node Group Attributes

Name	Description
Node Group	Click the Lookup icon and select Quick Find to select the Node Group you want to use. See "Use the Quick Find Window" on page 41 for more information about using Quick Find.
Ordering	Determines the priority order for those nodes that appear in multiple Node Groups. The lower the number, the higher the priority. For example, **1** is the highest priority. If a node is in multiple Node Groups and more than one of those Node Groups have been specified in an incident configuration, only the incident configuration with the highest priority will be applied to the node.
Enable	Use this attribute to temporarily disable an incident's suppression settings: **Enable** □ = Temporarily disable the selected configuration. **Enable** ☑ = Enable the selected configuration.

Configure Incident Suppression Settings for a Node Group (Syslog Message) (HP ArcSight)

Note: Node Settings override any other Suppression settings for this incident, except those configured on the Interface Settings tab.

NNMi enables you to suppress a specified incident configuration based on the Source Node's participation in a Node Group.

Note: You can also suppress the incident configuration based on the Source Object's participation in an Interface Group. See "Configure Incident Suppression Settings for an Interface Group (Syslog Message)(HP ArcSight)" on page 955 for more information.

Tip: See "Create Node Groups" on page 295for more information about Node Groups.

For information about each Node Settings tab:

To suppress an incident configuration based on a Node Group:

1. Navigate to the **Syslog Message Configuration** form:
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Expand the **Incidents** folder.
   c. Select **Syslog Message Configurations**.
   d. Do one of the following:
      i. To create an incident configuration, click the New icon, and continue.
      ii. To edit an incident configuration, select a row, click the Open icon, and continue.

iii. To delete an incident configuration, select a row, and click the ✖ Delete icon.

2. Select the **Node Settings** tab.

3. Make sure you configure the basic Node Setting behavior. See "Configure Node Settings for a Syslog Message Incident (HP ArcSight)" on page 991 for more information.

5. Select the **Suppression** tab.

6. Configure the desired Suppression behavior (see table).

7. Click **Save and Close** to save your changes and return to the previous form.

Node Settings Suppression Attributes

Name	Description
Enable	Use this attribute to temporarily disable an incident's suppression settings: **Enable** □ = Temporarily disable the selected configuration. **Enable** ☑ = Enable the selected configuration.

Payload Filter	The Payload Filter Editor enables you to create expressions that further refine the filters used to select the incidents to be suppressed, enriched, or dampened. Make sure to design any complex Payload Filters offline as a Boolean expression first. This method can help to minimize errors when entering your expressions using the Payload Filter editor.
	When creating a Payload Filter, note the following:
	• Payload Filter expressions for the `like` and `not like` operators use the syntax defined for java regular expressions (java.util.regex Pattern class).
	• You must use a `ciaName` that already exists in the trap or event you are configuring.
	• Each set of expressions associated with a Boolean Operator is treated as if it were enclosed in parentheses and evaluated together.
	• View the expression displayed under **Filter String** to see the logic of the expression as it is created.
	• The `AND` and `OR` Boolean Operators must contain at least two expressions as shown in the example below.
	The following example filters incidents on voltage state:
	`AND`
	    `ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7`
	    `ciaValue = 5`
	NNMi evaluates the expression above as follows:
	`(ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7 AND ciaValue = 5)`
	NNMi finds all incidents with a varbind `.1.3.6.1.4.1.9.9.13.1.2.1.7` value of **5**.
	• The placement of your cursor and the subsequent text that is selected is important

Node Settings Suppression Attributes , continued

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td></td><td>when performing operations using the Payload Filter Editor. For example, you append to, replace, or change the indentation of the expression that is selected.
• The placement of your cursor and the subsequent text that is selected is especially important when adding your Boolean operators.
• You can include more than one varbind in the same Payload Filter expression as shown in the following example:
<code>((ciaName like \Q.1.3.6.1.4.1.9.9\E.* AND ciaValue = 25) AND</code>
<code>(ciaName like \Q.1.3.6.1.2.1.2.2.1.1.3\E.* AND ciaValue = 3))</code>
In this example, a given trap must meet each of the following criteria:
▪ Contain a varbind whose Object Identifier (OID) matches the regular expression <code>\Q.1.3.6.1.4.1.9.9\E.*</code> and has a value of <code>25</code>.
▪ Contain a varbind whose OID matches the regular expression <code>\Q.1.3.6.1.2.1.2.2.1.1.3\E.*</code> and has a value of <code>3</code>.
<b>Payload Filter Editor Components</b>
<table>
<tr><th>Attribute</th><th>Description</th></tr>
<tr><td>Attribute</td><td>The attribute name on which NNMi searches. Filterable attributes include the following:
• ciaName
• ciaValue</td></tr>
<tr><td>Operator</td><td>Valid operators are described below.
• <b>=</b> Finds all values equal to the value specified. Click here for an example.
Example: <code>ciaName=.1.3.6.1.4.1.9.9.13.1.2.1.7</code> matches any incident that contains a varbind with the name value of <b>.1.3.6.1.4.1.9.9.13.1.2.1.7</b>.
• <b>!=</b> Finds all values not equal to the value specified. Click here for an example.
Example: <code>ciaName! = .1.3.6.1.4.1.9.9.13.1.2.1.7</code> matches any incident that contains a varbind with a name value other than <b>1.3.6.1.4.1.9.9.13.1.2.1.7</b>.
• <b><</b> Finds all values less than the value specified. Click here for an example.
Example: <code>ciaValue < 6</code> matches any incident that contains a varbind with a value less than <b>6</b>.
• <b><=</b> Finds all values less than or equal to the value specified. Click here for</td></tr>
</table></td></tr>
</table>

Node Settings Suppression Attributes , continued

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td></td><td>

Payload Filter Editor Components, continued

<table>
<tr><th>Attribute</th><th>Description</th></tr>
<tr><td></td><td>

an example.

Example: `ciaValue <= 6` matches any incident that contains a varbind with a value less than or equal to **6**.

- **>** Finds all values greater than the value specified. Click here for an example.

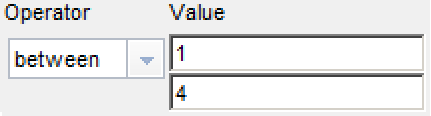
  Example: `ciaValue > 4` matches any incident that contains a varbind with a value greater than **4**.

- **>=** Finds all values greater than or equal to the value specified. Click here for an example.

  Example: `ciaValue >= 4` matches any incident that contains a varbind with values greater than or equal to **4**.

- **between** Finds all values equal to and between the two values specified. Click here for an example.

  Example: `ciaValue between`

  Operator: between
  Value: 1
  4

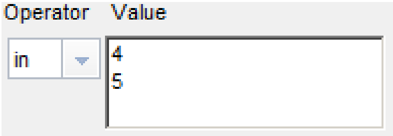
  matches any incident that contains a varbind value equal to or greater than **1** and equal to or less than **4**.

  Note: As shown in the example, each value must be entered on a separate line.

- **in** Finds any match to at least one value in a list of values. Click here for an example.

  Example:

  `ciaValue in`

  Operator: in
  Value: 4
  5

  matches any incident with a varbind value of either **4** or **5**.

  Note: As shown in the example, each value must be entered on a separate line.

  NNMi displays the list of attributes using comma-separated values

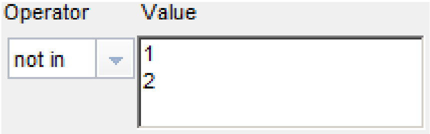
</td></tr>
</table>

</td></tr>
</table>

Node Settings Suppression Attributes , continued

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td></td><td>
<b>Payload Filter Editor Components, continued</b>
<table>
<tr><th>Attribute</th><th>Description</th></tr>
<tr><td></td><td>
<p>enclosed in parentheses, for example (<b>4, 5</b>). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line.</p>
<ul>
<li><p><b>is not null</b> Finds all non-blank values. Click here for an example.</p>
<p>Example: <code>ciaValue is not null</code> matches any incident with a varbind that contains a value.</p></li>
<li><p><b>is null</b> Finds all blank values. Click here for an example.</p>
<p>Example: <code>ciaValue is null</code> matches any incident with a varbind that does not contain a value.</p></li>
<li><p><b>like</b> Finds matches using the syntax defined for java regular expressions. See the Pattern (Java Platform SE6) API documentation at : <code>http://download.oracle.com/javase/6/docs/api/java/util/regex/Pattern.html</code> for more information. Click here for more information.</p>
<p>The period asterisk (.*) characters mean <i>any number of characters of any type at this location</i>.</p>
<p>The period (.) character means <i>any single character of any type at this location</i>.</p>
<p><b>Note</b>: To include literal string values in the Value attribute, enclose the string value in \Q&lt;<i>literal_value</i>&gt;\E as shown in the Examples listed below.</p>
<p>Example:</p>
<p><code>ciaName like \Q.1.3.6.1.4.1.9.9\E.*</code> matches any incident that contains a varbind name value that begins with 1.3.6.1.4.1.9.9 and (optionally) ends with any number of characters.</p>
<p><code>ciaValue like .*Chicago.*</code> finds all traps or events that contain a varbind value that includes the string <b>Chicago</b>.</p></li>
<li><p><b>not between</b> Finds all values except those between the two values specified. Click here for an example.</p>
<p>Example: <code>ciaValue not between 5 8</code> matches an incident that contains a varbind with the values less than <b>5</b> or greater than <b>8</b> .</p></li>
<li><p><b>not in</b> Finds all values except those included in the list of values. Click here for an example.</p>
<p>Example:</p></li>
</ul>
</td></tr>
</table>
</td></tr>
</table>

Node Settings Suppression Attributes , continued

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td></td><td>
<b>Payload Filter Editor Components, continued</b>
<table>
<tr><th>Attribute</th><th>Description</th></tr>
<tr><td></td><td>
<code>ciaValue not in</code>

Operator: not in
Value: 1
2
matches any incident that contains a varbind with values other than <b>1</b> and <b>2</b>.
<b>Note</b>: As shown in the example, each value must be entered on a separate line.
NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example, (<b>1, 2</b>). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line.
• <b>not like</b> Finds all that do not have the values specified using the syntax defined for Java regular expressions. See the Pattern (Java Platform SE6) API documentation at: <code>http://download.oracle.com/javase/6/docs/api/java/util/regex/Pattern.html</code> for more information. Click here for an example.
The period asterisk (.*) characters mean <i>any number of characters of any type at this location</i>.
The period (.) character means <i>any single character of any type at this location</i>.
<b>Note</b>: To include literal string values in the Value attribute, enclose the string value in \Q&lt;<i>literal_value</i>&gt;\E as shown in the Examples listed below.
Example:
<code>ciaName not like \Q.1.3.6.1.4.1.9.9\E.*</code> matches any incident that contains a varbind name value that does not begin with <b>.1.3.6.1.4.1.9.9</b> and (optionally) ends with any number of characters.
<code>ciaValue not like .*Chicago.*</code> finds all traps or events that do not contain a varbind value that includes the string <b>Chicago</b>.
</td></tr>
<tr><td>Value</td><td>The value for which you want NNMi to search.
Note the following:</td></tr>
</table>
</td></tr>
</table>

Node Settings Suppression Attributes , continued

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td></td><td>
<b>Payload Filter Editor Components, continued</b>
<table>
<tr><th>Attribute</th><th>Description</th></tr>
<tr><td></td><td>
<ul>
<li>The values you enter are case sensitive.</li>
<li>NNMi displays a variable number of value fields depending on the Operator selected. For example, the <code>between</code> Operator causes two value fields to be displayed.</li>
<li>The <code>between, in</code> and <code>not in</code> operators require that each value be entered on a separate line.</li>
</ul>
</td></tr>
</table>
<b>Payload Filter Editor Buttons</b>
<table>
<tr><th>Button</th><th>Description</th></tr>
<tr><td>Append</td><td>Appends the current expression (Attribute, Operator,and Value) to the selected expression already included in the filter string.</td></tr>
<tr><td>Insert</td><td>Inserts the current expression (Attribute, Operator,and Value) in front of the cursor location within the Filter String.</td></tr>
<tr><td>Replace</td><td>Replaces the selected expression with the expression displayed in the Attribute, Operator, and Value fields.</td></tr>
<tr><td>AND</td><td>Inserts the AND Boolean Operator in the selected cursor location.
<b>Note</b>: View the expression displayed under <b>Filter String</b> to see the logic of the expression as it is created.</td></tr>
<tr><td>OR</td><td>Inserts the OR Boolean Operator in the current cursor location.
<b>Note</b>: View the expression displayed under <b>Filter String</b> to see the logic of the expression as it is created.</td></tr>
<tr><td>NOT</td><td>Can be used in any part of the Filter String to specify that NNMi should exclude interfaces with values that pass the expression that immediately follows the <code>NOT</code>.
For example, when evaluating the following expression, NNMi includes interfaces with (interface description) <code>ifDesc</code> containing <b>VLAN</b>, and excludes any Interfaces that have <b>VLAN10</b> for the (interface name) <code>ifName</code> value:
<code>(ifDesc like VLAN AND NOT (ifName=VLAN10))</code>
<b>Note</b>: View the expression displayed under Filter String to see the logic of the expression as it is created .</td></tr>
</table>
</td></tr>
</table>

Node Settings Suppression Attributes , continued

<table>
<tr><th>Nam e</th><th>Description</th></tr>
<tr><td></td><td>
<b>Payload Filter Editor Buttons, continued</b>
<table>
<tr><th>Button</th><th>Description</th></tr>
<tr><td>EXISTS</td><td>Used for filters that include Capabilities or Custom Attribute names and values in the Filer String.
Indicates that you want NNMi to consider interfaces that do not have any Capabilities or Custom Attributes when evaluating the Filter String.
<b>Note</b>: If you include Capabilities or Custom Attribute names and values in the Filer String, but do not use <b>EXISTS</b> or <b>NOT EXISTS</b>, NNMi excludes from its search interfaces that do not include Capabilities or Custom Attributes.
For example, when evaluating the following Filter String, NNMi includes interfaces with (interface description) <code>ifDesc</code> containing <b>VLAN</b>, as well as any Interfaces Custom Attribute Role value is <b>LAN Connection to Oracle Server</b>:
<code>(ifDesc like VLAN OR EXISTS((customAttrName=Role AND customAttrValue=LAN Connection to Oracle Server)))</code>
<b>Note</b>: View the expression displayed under Filter String to see the logic of the expression as it is created.</td></tr>
<tr><td>NOT EXISTS</td><td>Used for filters that include Capabilities or Custom Attribute names and values in the Filer String. Indicates that you want NNMi to consider interfaces that do not have any Capabilities or Custom Attributes when evaluating the Filter String, but exclude the interfaces that match the expression that follows the <b>NOT EXISTS</b>.
<b>Note</b>: If you include Capabilities or Custom Attribute names and values in the Filer String, but do not use <b>EXISTS</b> or <b>NOT EXISTS</b>, NNMi excludes from its search interfaces that do not include Capabilities or Custom Attributes.
For example, when evaluating the following expression, NNMi includes interfaces with (interface description) <code>ifDesc</code> containing <b>VLAN</b>, and excludes any Interfaces that have the Custom Attribute Role and that Role value is <b>LAN Connection to Oracle Server</b>:
<code>(ifDesc like VLAN OR NOT EXISTS ((customAttrName=Role AND customAttrValue=LAN Connection to Oracle Server)))</code>
<b>Note</b>: View the expression displayed under Filter String to see the logic of the expression as it is created.</td></tr>
</table>
</td></tr>
</table>

Node Settings Suppression Attributes , continued

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td></td><td>
<b>Payload Filter Editor Buttons, continued</b>
<table>
<tr><th>Button</th><th>Description</th></tr>
<tr><td>Delete</td><td>Deletes the selected expression.
<b>Note</b>: If the Boolean Operator is selected, the Payload Filter Editor deletes all expressions associated with the Boolean Operator.</td></tr>
</table>
</td></tr>
</table>

Configure Incident Enrichment Settings for a Node Group (Syslog Message) (HP ArcSight)

Note: Node Settings override any other Enrichment settings for this incident, except those configured on the Interface Settings tab.

NNMi enables you to enhanced a specified incident configuration based on the Source Node's participation in a Node Group. The types of items you can fine tune and enhance for a selected incident configuration, include:

- Category
- Family
- Severity
- Priority
- Correlation Nature
- Message
- Assigned To

Note: You can also enhance the incident configuration based on the Source Object's participation in an Interface Group. See "Configure Incident Enrichment Settings for an Interface Group (Syslog Message)(HP ArcSight)" on page 963 for more information.

Tip: See "Create Node Groups" on page 295 for more information about Node Groups.

For information about each Node Settings tab:

For information about each Enrichment tab:

To configure Enrichment settings for a Node Group:

1. Navigate to the **Syslog Message Configuration** form:
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Select the **Incidents** folder.
   c. Select **Syslog Message Configurations**.
   d. Do one of the following:

   i. To create an incident configuration, click the ✱ New icon, and continue.

   ii. To edit an incident configuration, select a row, click the 📂 Open icon, and continue.

   iii. To delete an incident configuration, select a row, and click the ✖ Delete icon.

2. Select the **Node Settings** tab.
3. Do one of the following:
   a. To create a new configuration, click the ✱ New icon.

   b. To edit an existing configuration, select a row, click the 📂 Open icon, and continue.
4. Make sure you configure the basic Node Setting behavior. See "Configure Node Settings for a Syslog Message Incident (HP ArcSight)" on page 991 for more information.
5. Select the **Enrichment** tab.
6. Do one of the following:

   a. To create an Enrichment configuration, click the ✱ New icon and continue.

   b. To edit an Enrichment configuration, select a row, click the 📂 Open icon, and continue.

   c. To delete an Enrichment configuration, select a row and click the ✖ Delete icon.
7. Configure the desired Enrichment behavior (see the "Enrich Configuration Attributes" table)
8. Click 💾 **Save and Close** to save your changes and return to the previous form.

Node Settings Enrichment Configuration Attributes

Name	Description
Category	Use the Category attribute to customize the category for this incident configuration. Possible values include: • Accounting • Application Status • Configuration • Fault • Performance • Security • Status See "Specify Category and Family Attribute Values for Organizing Your Incidents (Syslog Message) (HP ArcSight)" on page 941 for more information.
Family	Use the Family attribute to customize the Family for this incident configuration. Select from the drop-down list or create a new value. For example, some of the values provided by NNMi include: • Address

Node Settings Enrichment Configuration Attributes , continued

Name	Description
	• Aggregated Port (Interfaces using **Link Aggregation**[1] protocol. See Interface Form: Link Aggregation tab.) • Card • Connection • Correlation • Interface • Node See "Specify Category and Family Attribute Values for Organizing Your Incidents (Syslog Message) (HP ArcSight)" on page 941 for more information.
Severity	The incident Severity represents the seriousness calculated for the incident. Use the Severity attribute to specify the severity that should be assigned to the incident you are configuring. Possible values are described below: **Normal** - Indicates there are no known problems related to the associated object. This Severity is meant to be informational. Generally, no action is needed for these incidents. **Warning** - Indicates there might be a problem related to the associated object. **Minor** - Indicates NNMi has detected problems related to the associated object that require further investigation. **Major** - Indicates NNMi has detected problems related to the associated object to be resolved before they become critical. **Critical** - Indicates NNMi has detected problems related to the associated object that require immediate attention.
Priority	Used to communicate the urgency of resolving the selected incident. You control this value. NNMi sets this value to null by default. The lower the number the higher the priority. Possible values are: 5 **None** 4 **Low**

[1] Protocols used on Switches to configure multiple Interfaces (Aggregation Member Interfaces) to function as if they were one (an Aggregator Interface). When two Aggregator Interfaces establish a connection, that connection is an Aggregator Layer 2 Connection. The Aggregator Layer 2 Connection appears on Layer 2 Neighbor View maps as a thick line with an Interface icon at each end (representing the Aggregator Interface). When you double-click the thick line, it converts into multiple thin lines representing the participating Aggregation Member Layer 2 Connections with their Aggregation Member Interfaces at each end of the lines.

Node Settings Enrichment Configuration Attributes , continued

Name	Description
	3 **Medium** 2 **High** 1 **Top** **Note**: The icons are displayed only in table views.
Correlation Nature	Use the Correlation Nature to customize the Correlation Nature for this incident configuration. Possible values include: • Info • None • Root Cause • Secondary Root Cause • Symptom • Stream Correlation • Service Impact • Dedup Stream Correlation • Rate Stream Correlation See Incident Form: General Tab for more information.
Message Format	When configuring an incident, specify how the incident message appears in the incident view. The string you specify in the Message Format attribute is visible in an incident view. **Note**: The incident message limit is 1024 characters. If you exceed this limit, NNMi truncates the value starting from the right. You can use any combination of default and custom attributes: "Valid Parameters for Configuring Incident Messages (Syslog Message) (HP ArcSight)" on page 946 "Include Custom Incident Attributes in Your Message Format (Syslog Message) (HP ArcSight)" on page 953
Assigned To	Use to specify the owner of any incident generated for this incident configuration. Click the Lookup icon and select Quick Find to select a valid user name. **Note**: You can also begin to type a valid user name and use the auto-complete feature to select the user name of interest.
Description	Use the Description attribute to provide additional information that you want to note about the current enhancement configuration. This description applies only to the enhancement configuration and does not appear when NNMi displays any

Node Settings Enrichment Configuration Attributes , continued

Name	Description
	associated incident. Type a maximum of 1024 characters. Alpha-numeric and special characters (~ ! @ # $ % ^ & * ( ) _+ -) are permitted.

Configure Custom Incident Attributes to Enrich an Incident Configuration (Node Settings) (Syslog Message) (HP ArcSight)

The Custom Incident Attributes (CIAs) tab enables you to specify additional CIAs to be saved with an incoming incident. The persisted data might then be used as an argument to an action for the incident.

When creating a CIA for an incident configuration, you can specify any of the following values:

- Custom Attribute on the source node
- Custom Attribute on the interface (source object)

You also specify the Custom Incident Attribute name that will store this information.

For information about each Enrichment tab:

To create a Custom Incident Attribute to enrich an incident configuration:

1. Navigate to the **Syslog Message Configuration** form:
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Expand the **Incidents** folder.
   c. Select **Syslog Message Configurations**.
   d. Do one of the following:
      i. To create an incident configuration, click the ✳ New icon, and continue.
      ii. To edit an incident configuration, select a row, click the 📂 Open icon, and continue.
      iii. To delete an incident configuration, select a row, and click the ✖ Delete icon.
2. Select the **Node Settings** tab.
3. Do one of the following:
   a. To create a new configuration, click the ✳ New icon.
   b. To edit an existing configuration, select a row, click the 📂 Open icon, and continue.
4. Make sure you configure the basic Node Setting behavior. See "Configure Node Settings for a Syslog Message Incident (HP ArcSight)" on page 991 for more information.
5. Select the **Enrichment** tab.
6. Do one of the following:

   a. To create an Enrichment configuration, click the New icon, and continue.

   b. To edit an Enrichment configuration,select a row, click the Open icon, and continue.

   c. To delete an Enrichment configuration, select a row and click the Delete icon.

7. Make sure the Enrichment settings are configure. See "Configure Incident Enrichment Settings for a Node Group (Syslog Message) (HP ArcSight)" on page 1000 for more information.

8. Navigate to the **Custom Incident Attributes** tab.

9. Do one of the following:

   a. To create a Custom Incident Attribute, click the New icon, and continue.

   b. To edit a Custom Incident Attribute, select a row, click the Open icon, and continue.

   c. To delete a Custom Incident Attribute, select a row and click the Delete icon.

10. Configure the Custom Incident Attribute (see the "Custom Incident Attribute" table).

11. Click **Save and Close** to save your changes and return to the previous form.

Custom Incident Attribute

Name	Description
Custom Incident Attribute Name	Name used to identify the Custom Incident Attribute you are configuring. The name limit is 255 characters. Alpha-numeric characters are permitted. No spaces or special characters (~ ! @ # $ % ^ & * ( ) _+ -) are permitted. **Note**: Make sure to note this name if you plan to filter on the value using the **Payload Filter** tab. See "Configure a Payload Filter to Enrich an Incident Configuration (Node Settings) (Syslog Message) (HP ArcSight)" below for more information.
Type	Specifies whether you are using a Custom Attribute on a node or a Custom Attribute on an interface. Possible values are: • Node Custom Attribute • Interface Custom Attribute
Custom Attribute Name	Used to determine the value to be assigned to the Custom Incident Attribute you are configuring. Enter either of the following: • Name of the Custom Attribute on the source node • Name of the Custom Attribute on the interface (source object)

Configure a Payload Filter to Enrich an Incident Configuration (Node Settings) (Syslog Message) (HP ArcSight)

The Payload Filter Editor enables you to create expressions that further refine the filters used to select the incidents to be enriched. Make sure to design any complex Payload Filters offline as a Boolean expression first. This method can help to minimize errors when entering your expressions using the Payload Filter editor.

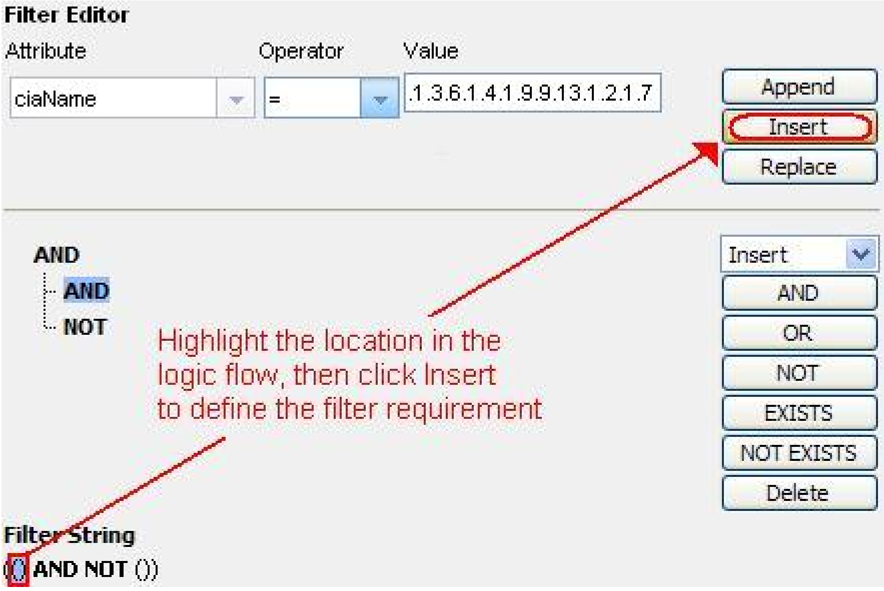
To create a Payload Filter expression:

1. Navigate to the **Syslog Message Configuration** form:
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Expand the **Incidents** folder.
   c. Select **Syslog Message Configurations**.
   d. Do one of the following:
      i. To create an incident configuration, click the * New icon, and continue.
      ii. To edit an incident configuration, select a row, click the Open icon, and continue.
      iii. To delete an incident configuration, select a row, and click the ✕ Delete icon.
2. Select the **Node Settings** tab.
3. Do one of the following:
   a. To create a new configuration, click the * New icon.
   b. To edit an existing configuration, double-click the row representing the configuration you want to edit.
4. Make sure you configure the basic Node Setting behavior. See "Configure Node Settings for a Syslog Message Incident (HP ArcSight)" on page 991 for more information.
5. Select the **Enrichment** tab.
6. Do one of the following:
   a. To create an Enrichment configuration, click the * New icon, and continue.
   b. To edit an Enrichment configuration, select a row, click the Open icon, and continue.
   c. To delete an Enrichment configuration, select a row and click the ✕ Delete icon.
7. Make sure you configure the Enrichment settings. See "Configure Incident Enrichment Settings for a Node Group (Syslog Message) (HP ArcSight)" on page 1000 for more information.
8. Select the **Payload Filter** tab.
9. Define your Payload Filter (see table). Also see "Guidelines for Creating a Payload Filter".
   a. Plan out the logic needed for your Filter String.
   b. Use the buttons on the bottom half of the Additional Filters Editor to establish the logic structure.

      For example, to establish the following structure, select **Insert**, then click **AND**, then **NOT**, and then **AND** a second time:

      ```
      (( ) AND NOT ( ))
      ```
   c. Now place your cursor in a location within the displayed Filter String, and use the top half of the filter editor to define the parameters of the highlighted filter requirement.

For example, select a set of parentheses and use the Insert button to specify the filter requirement within those parentheses:



Highlight the location in the logic flow, then click Insert to define the filter requirement

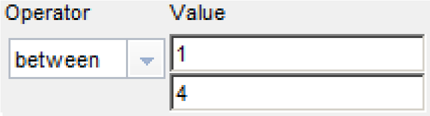
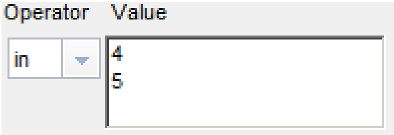
10. Click **Save and Close**.

11. Click **Save and Close** to save your changes and return to the previous form.

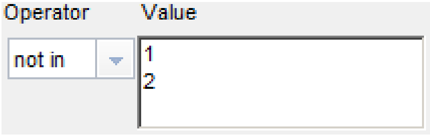
Payload Filter Editor Components

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: • ciaName • ciaValue
Operator	Valid operators are described below. • **=** Finds all values equal to the value specified. Click here for an example. Example: `ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7` matches any incident that contains a varbind with the name value **.1.3.6.1.4.1.9.9.13.1.2.1.7**. • **!=** Finds all values not equal to the value specified. Click here for an example. Example: `ciaName != .1.3.6.1.4.1.9.9.13.1.2.1.7` matches any incident that contains a varbind with the name value other than **1.3.6.1.4.1.9.9.13.1.2.1.7**. • **<** Finds all values less than the value specified. Click here for an example. Example: `ciaValue < 6` matches any incident with a varbind value less than **6**. • **<=** Finds all values less than or equal to the value specified. Click here for an example. Example: `ciaValue <= 6` matches any incident with a varbind value less than

Payload Filter Editor Components, continued

Attribute	Description
	or equal to **6**. - **>** Finds all values greater than the value specified. Click here for an example. Example: `ciaValue > 4` matches any incident with a varbind value greater than **4**. - **>=** Finds all values greater than or equal to the value specified. Click here for an example. Example: `ciaValue >= 4` matches any incident that contains a varbind with values greater than or equal to **4**. - **between** Finds all traps or events that include a varbind with a value equal to and between the two values specified. Click here for an example. Example: `ciaValue between`  Operator between Value 1 4 matches any incident that contains a varbind value equal to or greater than **1** and equal to or less than **4**. **Note**: As shown in the example, each value must be entered on a separate line. - **in** Finds any match to at least one value in a list of values. Click here for an example. Example: `ciaValue in`  Operator in Value 4 5 matches any incident with a varbind value of either **4** or **5**. **Note**: As shown in the example, each value must be entered on a separate line. NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example (**4, 5**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line. - **is not null** Finds all non-blank values. Click here for an example. Example: `ciaValue is not null` matches any incident with a varbind that contains a value. - **is null** Finds all blank values. Click here for an example. Example: `ciaValue is null` matches any incident with a varbind that does not have a value.

Payload Filter Editor Components, continued

Attribute	Description
	• **like** Finds matches using wildcard characters. Click here for more information about using wildcard characters. The period asterisk (.*) characters mean *any number of characters of any type at this location*. The period (.) character means *any single character of any type at this location*. **Note**: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below. Examples: `ciaName like  \Q.1.3.6.1.4.1.9.9\E.*` finds all traps or events that contain varbind names that begin with **.1.3.6.1.4.1.9.9** and (optionally) end with any number of characters. `ciaValue like .*Chicago.*` finds all traps or events that contain a varbind value that includes the string **Chicago**. • **not between** Finds all values except those between the two values specified. Click here for an example. Example: `ciaValue not between 5 8` matches an incident that contains a varbind with the values less than **5** or greater than **8**. • **not in** Finds all values except those included in the list of values. Click here for an example. Example: `ciaValue not in`  Operator not in Value 1 2 matches any incident that contains a varbind with values other than **1** and **2**. **Note**: As shown in the example, each value must be entered on a separate line. NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example, (**1, 2**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line. • **not like** Finds all that do not have the values specified (using wildcard strings). Click here for an example. The period asterisk (.*) characters mean *any number of characters of any type at this location*. The period (.) character means *any single character of any type at this location*. **Note**: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below.

Payload Filter Editor Components, continued

Attribute	Description
	Example: `ciaName not like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that does not begin with **.1.3.6.1.4.1.9.9**. `ciaValue not like .*Chicago.*` finds all traps or events that do not contain a varbind value that includes the string **Chicago**.
Value	The value for which you want NNMi to search. Note the following: • The values you enter are case sensitive. • NNMi displays a variable number of value fields depending on the Operator selected. For example, the `between` Operator causes two value fields to be displayed. • The `between, in` and `not in` operators require that each value be entered on a separate line.

Additional Filters Editor Buttons

Button	Description
Append	Appends the current expression (Attribute, Operator,and Value) to the selected expression already included in the Filter String.
Insert	Inserts the current expression (Attribute, Operator,and Value) in front of the cursor location within the Filter String.
Replace	Replaces the selected expression with the expression displayed in the Attribute, Operator, and Value fields.
AND	Appends, inserts, or replaces the AND Boolean Operator in the selected cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
OR	Appends, Inserts, or replaces the OR Boolean Operator in the current cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
NOT	Can be used in any part of the Filter String to specify that NNMi should exclude interfaces with values that pass the expression that immediately follows the `NOT`. For example, when evaluating the following expression, NNMi includes interfaces with (interface description) `ifDesc` containing **VLAN**, and excludes any Interfaces that have **VLAN10** for the (interface name) `ifName` value:

Additional Filters Editor Buttons, continued

Button	Description
	`(ifDesc like VLAN AND NOT (ifName=VLAN10))` **Note**: View the expression displayed under Filter String to see the logic of the expression as it is created .
EXISTS	Used for filters that include Capabilities or Custom Attribute names and values in the Filer String. Indicates that you want NNMi to consider interfaces that do not have any Capabilities or Custom Attributes when evaluating the Filter String. **Note**: If you include Capabilities or Custom Attribute names and values in the Filer String, but do not use **EXISTS** or **NOT EXISTS**, NNMi excludes from its search interfaces that do not include Capabilities or Custom Attributes. For example, when evaluating the following Filter String, NNMi includes interfaces with (interface description) `ifDesc` containing **VLAN**, as well as any Interfaces Custom Attribute Role value is **LAN Connection to Oracle Server**: `(ifDesc like VLAN OR EXISTS((customAttrName=Role AND customAttrValue=LAN Connection to Oracle Server)))` **Note**: View the expression displayed under Filter String to see the logic of the expression as it is created.
NOT EXISTS	Used for filters that include Capabilities or Custom Attribute names and values in the Filer String. Indicates that you want NNMi to consider interfaces that do not have any Capabilities or Custom Attributes when evaluating the Filter String, but exclude the interfaces that match the expression that follows the **NOT EXISTS**. Note: If you include Capabilities or Custom Attribute names and values in the Filer String, but do not use **EXISTS** or **NOT EXISTS**, NNMi excludes from its search interfaces that do not include Capabilities or Custom Attributes. For example, when evaluating the following expression, NNMi includes interfaces with (interface description) `ifDesc` containing **VLAN**, and excludes any Interfaces that have the Custom Attribute Role and that Role value is **LAN Connection to Oracle Server**: `(ifDesc like VLAN OR NOT EXISTS((customAttrName=Role AND customAttrValue=LAN Connection to Oracle Server)))` **Note**: View the expression displayed under Filter String to see the logic of the expression as it is created.
Delete	Deletes the selected expression. **Note**: If the Boolean Operator is selected, the Additional Filters Editor deletes all expressions associated with the Boolean Operator.

Configure Incident Dampening Settings for a Node Group (Syslog Message) (HP ArcSight)

Note: Node Settings override any other Dampening settings for this incident, except those configured on the Interface Settings tab.

NNMi enables you to delay the following for an incident configuration based on the Source Node's participation in a Node Group:

- Execution of Incident Actions
- Execution of Diagnostics (*HP Network Node Manager iSPI Network Engineering Toolset Software \ NNM iSPI NET*)
- Appearance within Incident views in the NNMi Console

Note: You can configure the Dampening settings based on the Source Object's participation in an Interface Group. See "Configure Incident Dampening Settings for an Interface Group (Syslog Message) (HP ArcSight)" on page 975 for more information.

Tip: See "Create Node Groups" on page 295 for more information about Node Groups.

For information about each Node Settings tab:

When using the Dampening configuration, note the following:

- NNMi initially assigns incidents with Dampening settings configured a Lifecycle State of DAMPENED.
- After the dampen interval has passed, NNMi changes the Lifecycle State to REGISTERED.

See About the Incident Lifecycle for more information about Lifecycle State.

To configure the Dampening settings based on a Node Group:

1. Navigate to the **Syslog Message Configuration** form:
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Expand the **Incidents** folder.
   c. Select **Syslog Message Configurations**.
   d. Do one of the following:
      i. To create an incident configuration, click the ✳ New icon, and continue.
      ii. To edit an incident configuration, select a row, click the 📂 Open icon, and continue.
      iii. To delete an incident configuration, select a row, and click the ✖ Delete icon.
2. Select the **Node Settings** tab.
3. Do one of the following:
   a. To create a new configuration, click the ✳ New icon.
   b. To edit an existing configuration, select a row, click the 📂 Open icon, and continue.

4. Make sure you configure the basic Node Setting behavior. See "Configure Node Settings for a Syslog Message Incident (HP ArcSight)" on page 991 for more information.
5. Select the **Dampen** tab.
6. Configure the desired Dampen behavior (see table).
7. Click **Save and Close** to save your changes and return to the previous form.

Node Settings Dampen Attributes

Name	Description
Enable	Use this attribute to temporarily disable an incident's Dampening settings: **Enable** □ = Temporarily disable the selected configuration. **Enable** ☑ = Enable the selected configuration.
Hour	Specifies the number of hours to be used for the dampen interval.
Minutes	Specifies the number of minutes to be used for the dampen interval. **Note:** The maximum dampen interval is 60 minutes.
Seconds	Specifies the number of seconds to be used for the dampen interval.

Payload Filter	The Payload Filter Editor enables you to create expressions that further refine the filters used to select the incidents to be suppressed, enriched, or dampened. Make sure to design any complex Payload Filters offline as a Boolean expression first. This method can help to minimize errors when entering your expressions using the Payload Filter editor.
	When creating a Payload Filter, note the following:
	• Payload Filter expressions for the `like` and `not like` operators use the syntax defined for java regular expressions (java.util.regex Pattern class).
	• You must use a `ciaName` that already exists in the trap or event you are configuring.
	• Each set of expressions associated with a Boolean Operator is treated as if it were enclosed in parentheses and evaluated together.
	• View the expression displayed under **Filter String** to see the logic of the expression as it is created.
	• The `AND` and `OR` Boolean Operators must contain at least two expressions as shown in the example below.
	The following example filters incidents on voltage state:
	`AND`
	`    ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7`
	`    ciaValue = 5`
	NNMi evaluates the expression above as follows:

Node Settings Dampen Attributes , continued

Name	Description

```
(ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7 AND ciaValue = 5)
```

NNMi finds all incidents with a varbind .1.3.6.1.4.1.9.9.13.1.2.1.7 value of **5**.

- The placement of your cursor and the subsequent text that is selected is important when performing operations using the Payload Filter Editor. For example, you append to, replace, or change the indentation of the expression that is selected.
- The placement of your cursor and the subsequent text that is selected is especially important when adding your Boolean operators.
- You can include more than one varbind in the same Payload Filter expression as shown in the following example:

```
((ciaName like \Q.1.3.6.1.4.1.9.9\E.* AND ciaValue = 25) AND
(ciaName like \Q.1.3.6.1.2.1.2.2.1.1.3\E.* AND ciaValue =
3))
```

In this example, a given trap must meet each of the following criteria:

- Contain a varbind whose Object Identifier (OID) matches the regular expression \Q.1.3.6.1.4.1.9.9\E.* and has a value of 25.
- Contain a varbind whose OID matches the regular expression \Q.1.3.6.1.2.1.2.2.1.1.3\E.* and has a value of 3.

Payload Filter Editor Components

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: • ciaName • ciaValue
Operator	Valid operators are described below. • **=** Finds all values equal to the value specified. Click here for an example. Example: ciaName=.1.3.6.1.4.1.9.9.13.1.2.1.7 matches any incident that contains a varbind with the name value of **.1.3.6.1.4.1.9.9.13.1.2.1.7**. • **!=** Finds all values not equal to the value specified. Click here for an example. Example: ciaName! = .1.3.6.1.4.1.9.9.13.1.2.1.7 matches any incident that contains a varbind with a name value other than **1.3.6.1.4.1.9.9.13.1.2.1.7**.

Node Settings Dampen Attributes , continued

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td></td><td>

Payload Filter Editor Components, continued

<table>
<tr><th>Attribute</th><th>Description</th></tr>
<tr><td></td><td>

- **<** Finds all values less than the value specified. Click here for an example.

  Example: `ciaValue < 6` matches any incident that contains a varbind with a value less than **6**.

- **<=** Finds all values less than or equal to the value specified. Click here for an example.

  Example: `ciaValue <= 6` matches any incident that contains a varbind with a value less than or equal to **6**.

- **>** Finds all values greater than the value specified. Click here for an example.

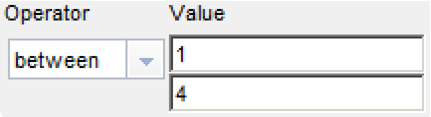
  Example: `ciaValue > 4` matches any incident that contains a varbind with a value greater than **4**.

- **>=** Finds all values greater than or equal to the value specified. Click here for an example.

  Example: `ciaValue >= 4` matches any incident that contains a varbind with values greater than or equal to **4**.

- **between** Finds all values equal to and between the two values specified. Click here for an example.

  Example: `ciaValue between`

  Operator: between
  Value: 1
  4

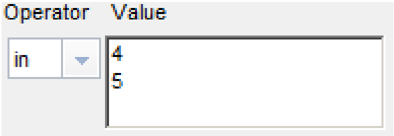
  matches any incident that contains a varbind value equal to or greater than **1** and equal to or less than **4**.

  Note: As shown in the example, each value must be entered on a separate line.

- **in** Finds any match to at least one value in a list of values. Click here for an example.

  Example:

  `ciaValue in`

  Operator: in
  Value: 4
  5

  matches any incident with a varbind value of either **4** or **5**.

</td></tr>
</table>

</td></tr>
</table>

Node Settings Dampen Attributes , continued

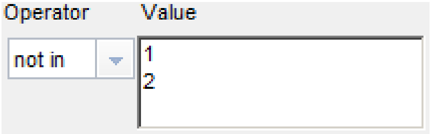
Name	Description
	Payload Filter Editor Components, continued

Attribute	Description
	Note: As shown in the example, each value must be entered on a separate line. NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example (**4, 5**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line. • **is not null** Finds all non-blank values. Click here for an example. Example: `ciaValue is not null` matches any incident with a varbind that contains a value. • **is null** Finds all blank values. Click here for an example. Example: `ciaValue is null` matches any incident with a varbind that does not contain a value. • **like** Finds matches using the syntax defined for java regular expressions. See the Pattern (Java Platform SE6) API documentation at : `http://download.oracle.com/javase/6/docs/api/java/util/regex/Pattern.html` for more information. Click here for more information. The period asterisk (.*) characters mean *any number of characters of any type at this location*. The period (.) character means *any single character of any type at this location*. **Note**: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below. Example: `ciaName like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that begins with 1.3.6.1.4.1.9.9 and (optionally) ends with any number of characters. `ciaValue like .*Chicago.*` finds all traps or events that contain a varbind value that includes the string **Chicago**. • **not between** Finds all values except those between the two values specified. Click here for an example. Example: `ciaValue not between 5 8` matches an incident that contains a varbind with the values less than **5** or greater than **8** .

Node Settings Dampen Attributes , continued

Name	Description

Payload Filter Editor Components, continued

Attribute	Description
	- **not in** Finds all values except those included in the list of values. Click here for an example. Example: `ciaValue not in`  Operator: not in Value: 1 2 matches any incident that contains a varbind with values other than **1** and **2**. **Note**: As shown in the example, each value must be entered on a separate line. NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example, (**1, 2**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line. - **not like** Finds all that do not have the values specified using the syntax defined for Java regular expressions. See the Pattern (Java Platform SE6) API documentation at: `http://download.oracle.com/javase/6/docs/api/java/util/regex/Pattern.html` for more information. Click here for an example. The period asterisk (.*) characters mean *any number of characters of any type at this location*. The period (.) character means *any single character of any type at this location*. **Note**: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below. Example: `ciaName not like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that does not begin with **.1.3.6.1.4.1.9.9** and (optionally) ends with any number of characters. `ciaValue not like .*Chicago.*` finds all traps or events that do not contain a varbind value that includes the string **Chicago**.

Node Settings Dampen Attributes , continued

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td></td><td>

Payload Filter Editor Components, continued

<table>
<tr><th>Attribute</th><th>Description</th></tr>
<tr><td>Value</td><td>The value for which you want NNMi to search.
Note the following:
• The values you enter are case sensitive.
• NNMi displays a variable number of value fields depending on the Operator selected. For example, the `between` Operator causes two value fields to be displayed.
• The `between, in` and `not in` operators require that each value be entered on a separate line.</td></tr>
</table>

Payload Filter Editor Buttons

<table>
<tr><th>Button</th><th>Description</th></tr>
<tr><td>Append</td><td>Appends the current expression (Attribute, Operator,and Value) to the selected expression already included in the filter string.</td></tr>
<tr><td>Insert</td><td>Inserts the current expression (Attribute, Operator,and Value) in front of the cursor location within the Filter String.</td></tr>
<tr><td>Replace</td><td>Replaces the selected expression with the expression displayed in the Attribute, Operator, and Value fields.</td></tr>
<tr><td>AND</td><td>Inserts the AND Boolean Operator in the selected cursor location.
Note: View the expression displayed under **Filter String** to see the logic of the expression as it is created.</td></tr>
<tr><td>OR</td><td>Inserts the OR Boolean Operator in the current cursor location.
Note: View the expression displayed under **Filter String** to see the logic of the expression as it is created.</td></tr>
<tr><td>NOT</td><td>Can be used in any part of the Filter String to specify that NNMi should exclude interfaces with values that pass the expression that immediately follows the `NOT`.
For example, when evaluating the following expression, NNMi includes interfaces with (interface description) `ifDesc` containing **VLAN**, and excludes any Interfaces that have **VLAN10** for the (interface name) `ifName` value:
`(ifDesc like VLAN AND NOT (ifName=VLAN10))`</td></tr>
</table>

</td></tr>
</table>

Node Settings Dampen Attributes , continued

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td></td><td>

Payload Filter Editor Buttons, continued

<table>
<tr><th>Button</th><th>Description</th></tr>
<tr><td></td><td><b>Note</b>: View the expression displayed under Filter String to see the logic of the expression as it is created .</td></tr>
<tr><td>EXISTS</td><td>Used for filters that include Capabilities or Custom Attribute names and values in the Filer String.
Indicates that you want NNMi to consider interfaces that do not have any Capabilities or Custom Attributes when evaluating the Filter String.
<b>Note</b>: If you include Capabilities or Custom Attribute names and values in the Filer String, but do not use <b>EXISTS</b> or <b>NOT EXISTS</b>, NNMi excludes from its search interfaces that do not include Capabilities or Custom Attributes.
For example, when evaluating the following Filter String, NNMi includes interfaces with (interface description) <code>ifDesc</code> containing <b>VLAN</b>, as well as any Interfaces Custom Attribute Role value is <b>LAN Connection to Oracle Server</b>:
<code>(ifDesc like VLAN OR EXISTS((customAttrName=Role AND customAttrValue=LAN Connection to Oracle Server)))</code>
<b>Note</b>: View the expression displayed under Filter String to see the logic of the expression as it is created.</td></tr>
<tr><td>NOT EXISTS</td><td>Used for filters that include Capabilities or Custom Attribute names and values in the Filer String. Indicates that you want NNMi to consider interfaces that do not have any Capabilities or Custom Attributes when evaluating the Filter String, but exclude the interfaces that match the expression that follows the <b>NOT EXISTS</b>.
<b>Note</b>: If you include Capabilities or Custom Attribute names and values in the Filer String, but do not use <b>EXISTS</b> or <b>NOT EXISTS</b>, NNMi excludes from its search interfaces that do not include Capabilities or Custom Attributes.
For example, when evaluating the following expression, NNMi includes interfaces with (interface description) <code>ifDesc</code> containing <b>VLAN</b>, and excludes any Interfaces that have the Custom Attribute Role and that Role value is <b>LAN Connection to Oracle Server</b>:
<code>(ifDesc like VLAN OR NOT EXISTS ((customAttrName=Role AND customAttrValue=LAN Connection to Oracle Server)))</code></td></tr>
</table>

</td></tr>
</table>

Node Settings Dampen Attributes , continued

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td></td><td>
<b>Payload Filter Editor Buttons, continued</b>
<table>
<tr><th>Button</th><th>Description</th></tr>
<tr><td></td><td><b>Note</b>: View the expression displayed under Filter String to see the logic of the expression as it is created.</td></tr>
<tr><td>Delete</td><td>Deletes the selected expression.
<b>Note</b>: If the Boolean Operator is selected, the Payload Filter Editor deletes all expressions associated with the Boolean Operator.</td></tr>
</table>
</td></tr>
</table>

Configure Incident Actions for a Node Group (Syslog Message) (HP ArcSight)

For information about each Node Settings tab:

Note: Node Settings override any other Actions settings for this incident, except those configured on the Interface Settings tab.

NNMi enables you to configure incident actions based on a Source Node's participation in a Node Group.

You can configure actions to automatically run at any point in the incident lifecycle. For example, you might want to configure an action to occur when an incident of the type you are configuring is generated (**Registered**). When an incident is generated, you might want to automatically open a trouble ticket or send email or page your network operator. After the incident is **Closed**, you might want to automatically close the trouble ticket.

Note: Your actions will not be executed until you enable the Actions configuration by either clicking Enable ☑ on the Actions tab or using the **Actions** → **Enable Configuration** option.

You can configure actions for incidents generated from SNMP traps, Syslog Messages (HP ArcSightonly), Remote NNM 6.x or 7.x events, and the NNMi Management Events. Any time an incident configuration changes, the action directory is rescanned and any Jython files are reloaded to the NNMi database. See "Lifecycle Transition Action Form (Syslog Message) (HP ArcSight)" on page 1064 for more information about the actions directory.

Tip: Copy any required Jython files to the NNMi actions directory before you configure an incident action. New or updated actions are loaded into NNMi only when an incident configuration is updated or created. See "Lifecycle Transition Action Form (Management Events)" on page 1207 for the location of the NNMi action directory.

When the action runs, output is logged to the `incidentActions.*.*.log` file. See "Verify that NNMi Services are Running" on page 85 for more information about log files and where they are located.

To configure an automatic action for an incident:

1. Navigate to the **Syslog Message Configuration** tab.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Select the **Incidents** folder.
   c. Select **Syslog Message Configurations**.
   d. Do one of the following:
      i. To create a new incident configuration, click the New icon.
      ii. To edit an existing incident configurationselect a row, click the Open icon, and continue.
2. Select the **Node Settings** tab.
3. Do one of the following:
   a. To create a new configuration, click the New icon.
   b. To edit an existing configuration, double-click the row representing the configuration you want to edit.
4. Make sure you configure the basic Node Setting behavior. See "Configure Node Settings for a Syslog Message Incident (HP ArcSight)" on page 991 for more information.
5. Select the **Actions** tab.
6. From the **Lifecycle Actions** table toolbar, do one of the following:
   - To create an Action configuration, click the New icon, and continue.
   - To edit an Action configuration, double-click the row representing the configuration you want to edit, and continue.
   - To delete an Action configuration, select a row, and click the Delete icon.
7. In the "Lifecycle Transition Action Form (Management Events)" on page 1207, provide the required information.
8. Click **Save and Close** to save your changes and return to the **Syslog Message Configuration** form.

   The next time the lifecycle changes, NNMi launches the action associated with the lifecycle for the incident of that type.

Configure a Payload Filter for an Incident Action (Node Settings) (Syslog Message) (HP ArcSight)

The Payload Filter Editor enables you to create expressions that further refine the filters used to select the incidents that cause the configured action to run. Make sure to design any complex Payload Filters offline as a Boolean expression first. This method can help to minimize errors when entering your expressions using the Payload Filter editor.

To create a Payload Filter expression:

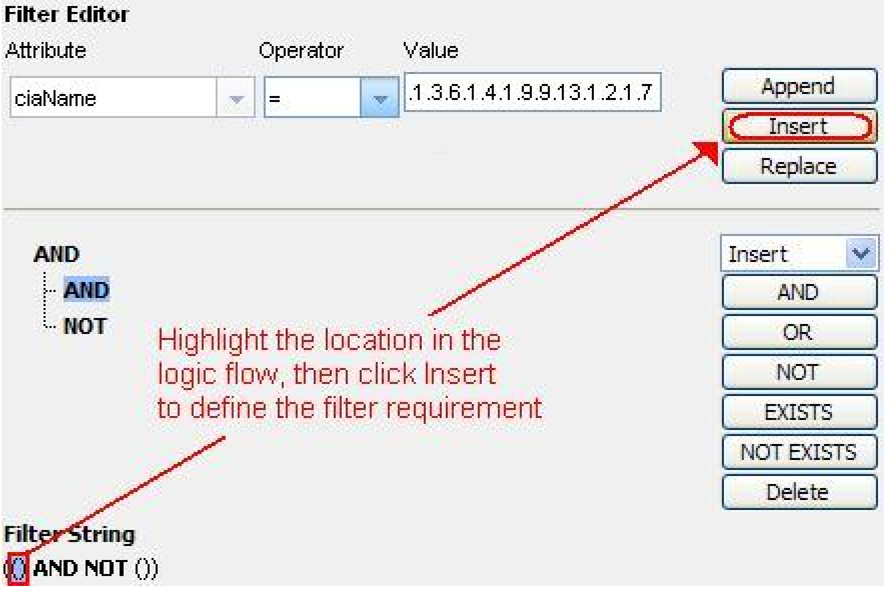
1. Navigate to the **Syslog Message Configuration** form:
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Expand the **Incidents** folder.
   c. Select **Syslog Message Configurations**.
   d. Do one of the following:
      i. To create an incident configuration, click the ✳ New icon, and continue.
      ii. To edit an incident configuration, select a row, click the 📂 Open icon, and continue.
      iii. To delete an incident configuration, select a row, and click the ✖ Delete icon.
2. Select the **Node Settings** tab.
3. Do one of the following:
   a. To create a new configuration, click the ✳ New icon.
   b. To edit an existing configuration, select a row, click the 📂 Open icon, and continue.
4. Make sure you configure the basic Node Setting behavior. See "Configure Node Settings for a Syslog Message Incident (HP ArcSight)" on page 991 for more information.
5. Select the **Actions** tab.
6. Do one of the following:
   a. To create an Action configuration, click the ✳ New icon, and continue.
   b. To edit an Action configuration, select a row, click the 📂 Open icon, and continue.
   c. To delete an Action configuration, select a row, and click the ✖ Delete icon.
7. Make sure the Action settings are configured. See "Configure Incident Actions for a Node Group (Syslog Message) (HP ArcSight)" on page 1020 for more information.
8. Select the **Payload Filter** tab.
9. Define your Payload Filter (see table). Also see "Guidelines for Creating a Payload Filter".
   a. Plan out the logic needed for your Filter String.
   b. Use the buttons on the bottom half of the Additional Filters Editor to establish the logic structure.

      For example, to establish the following structure, select **Insert**, then click **AND**, then **NOT**, and then **AND** a second time:

      ```
      (( ) AND NOT ( ))
      ```
   c. Now place your cursor in a location within the displayed Filter String, and use the top half of the filter editor to define the parameters of the highlighted filter requirement.

      For example, select a set of parentheses and use the Insert button to specify the filter

requirement within those parentheses:



Highlight the location in the logic flow, then click Insert to define the filter requirement

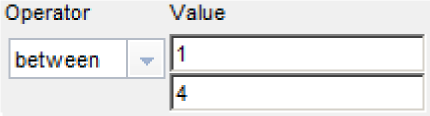
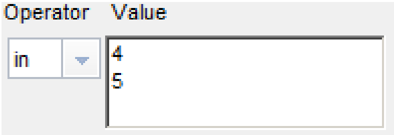
10. Click **Save and Close.**

11. Click **Save and Close** to save your changes and return to the previous form.

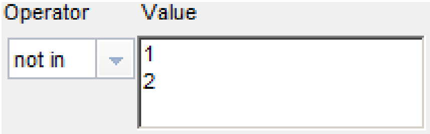
Payload Filter Editor Components

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: • ciaName • ciaValue
Operator	Valid operators are described below. • **=** Finds all values equal to the value specified. Click here for an example. Example: `ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7` matches any incident that contains a varbind with the name value **.1.3.6.1.4.1.9.9.13.1.2.1.7**. • **!=** Finds all values not equal to the value specified. Click here for an example. Example: `ciaName != .1.3.6.1.4.1.9.9.13.1.2.1.7` matches any incident that contains a varbind with the name value other than **1.3.6.1.4.1.9.9.13.1.2.1.7**. • **<** Finds all values less than the value specified. Click here for an example. Example: `ciaValue < 6` matches any incident with a varbind value less than **6**. • **<=** Finds all values less than or equal to the value specified. Click here for an example. Example: `ciaValue <= 6` matches any incident with a varbind value less than or equal to **6**.

Payload Filter Editor Components, continued

Attribute	Description
	• **>** Finds all values greater than the value specified. Click here for an example. Example: `ciaValue > 4` matches any incident with a varbind value greater than **4**. • **>=** Finds all values greater than or equal to the value specified. Click here for an example. Example: `ciaValue >= 4` matches any incident that contains a varbind with values greater than or equal to **4**. • **between** Finds all traps or events that include a varbind with a value equal to and between the two values specified. Click here for an example. Example: `ciaValue between`  Operator Value between 1 4 matches any incident that contains a varbind value equal to or greater than **1** and equal to or less than **4**. **Note**: As shown in the example, each value must be entered on a separate line. • **in** Finds any match to at least one value in a list of values. Click here for an example. Example: `ciaValue in`  Operator Value in 4 5 matches any incident with a varbind value of either **4** or **5**. **Note**: As shown in the example, each value must be entered on a separate line. NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example (**4, 5**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line. • **is not null** Finds all non-blank values. Click here for an example. Example: `ciaValue is not null` matches any incident with a varbind that contains a value. • **is null** Finds all blank values. Click here for an example. Example: `ciaValue is null` matches any incident with a varbind that does not have a value. • **like** Finds matches using wildcard characters. Click here for more information

Payload Filter Editor Components, continued

Attribute	Description
	about using wildcard characters. The period asterisk (.*) characters mean *any number of characters of any type at this location*. The period (.) character means *any single character of any type at this location*. **Note**: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below. Examples: `ciaName like  \Q.1.3.6.1.4.1.9.9\E.*` finds all traps or events that contain varbind names that begin with **.1.3.6.1.4.1.9.9** and (optionally) end with any number of characters. `ciaValue like .*Chicago.*` finds all traps or events that contain a varbind value that includes the string **Chicago**. • **not between** Finds all values except those between the two values specified. Click here for an example. Example: `ciaValue not between 5 8` matches an incident that contains a varbind with the values less than **5** or greater than **8**. • **not in** Finds all values except those included in the list of values. Click here for an example. Example: `ciaValue not in`  Operator Value not in 1 2 matches any incident that contains a varbind with values other than **1** and **2**. **Note**: As shown in the example, each value must be entered on a separate line. NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example, (**1, 2**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line. • **not like** Finds all that do not have the values specified (using wildcard strings). Click here for an example. The period asterisk (.*) characters mean *any number of characters of any type at this location*. The period (.) character means *any single character of any type at this location*. **Note**: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below. Example:

Payload Filter Editor Components, continued

Attribute	Description
	`ciaName not like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that does not begin with **.1.3.6.1.4.1.9.9**. `ciaValue not like .*Chicago.*` finds all traps or events that do not contain a varbind value that includes the string **Chicago**.
Value	The value for which you want NNMi to search. Note the following: • The values you enter are case sensitive. • NNMi displays a variable number of value fields depending on the Operator selected. For example, the `between` Operator causes two value fields to be displayed. • The `between, in` and `not in` operators require that each value be entered on a separate line.

Additional Filters Editor Buttons

Button	Description
Append	Appends the current expression (Attribute, Operator,and Value) to the selected expression already included in the Filter String.
Insert	Inserts the current expression (Attribute, Operator,and Value) in front of the cursor location within the Filter String.
Replace	Replaces the selected expression with the expression displayed in the Attribute, Operator, and Value fields.
AND	Appends, inserts, or replaces the AND Boolean Operator in the selected cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
OR	Appends, Inserts, or replaces the OR Boolean Operator in the current cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
NOT	Can be used in any part of the Filter String to specify that NNMi should exclude interfaces with values that pass the expression that immediately follows the `NOT`. For example, when evaluating the following expression, NNMi includes interfaces with (interface description) `ifDesc` containing **VLAN**, and excludes any Interfaces that have **VLAN10** for the (interface name) `ifName` value: `(ifDesc like VLAN AND NOT (ifName=VLAN10))`

Additional Filters Editor Buttons, continued

Button	Description
	Note: View the expression displayed under Filter String to see the logic of the expression as it is created .
EXISTS	Used for filters that include Capabilities or Custom Attribute names and values in the Filer String. Indicates that you want NNMi to consider interfaces that do not have any Capabilities or Custom Attributes when evaluating the Filter String. **Note**: If you include Capabilities or Custom Attribute names and values in the Filer String, but do not use **EXISTS** or **NOT EXISTS**, NNMi excludes from its search interfaces that do not include Capabilities or Custom Attributes. For example, when evaluating the following Filter String, NNMi includes interfaces with (interface description) `ifDesc` containing **VLAN**, as well as any Interfaces Custom Attribute Role value is **LAN Connection to Oracle Server**: `(ifDesc like VLAN OR EXISTS((customAttrName=Role AND customAttrValue=LAN Connection to Oracle Server)))` **Note**: View the expression displayed under Filter String to see the logic of the expression as it is created.
NOT EXISTS	Used for filters that include Capabilities or Custom Attribute names and values in the Filer String. Indicates that you want NNMi to consider interfaces that do not have any Capabilities or Custom Attributes when evaluating the Filter String, but exclude the interfaces that match the expression that follows the **NOT EXISTS**. Note: If you include Capabilities or Custom Attribute names and values in the Filer String, but do not use **EXISTS** or **NOT EXISTS**, NNMi excludes from its search interfaces that do not include Capabilities or Custom Attributes. For example, when evaluating the following expression, NNMi includes interfaces with (interface description) `ifDesc` containing **VLAN**, and excludes any Interfaces that have the Custom Attribute Role and that Role value is **LAN Connection to Oracle Server**: `(ifDesc like VLAN OR NOT EXISTS((customAttrName=Role AND customAttrValue=LAN Connection to Oracle Server)))` **Note**: View the expression displayed under Filter String to see the logic of the expression as it is created.
Delete	Deletes the selected expression. **Note**: If the Boolean Operator is selected, the Additional Filters Editor deletes all expressions associated with the Boolean Operator.

Configure Diagnostics Selections for a Node Group (Syslog Message) (HP ArcSight)

For information about each Node Settings tab: .

Note: Node Settings override any other Diagnostics Selections settings for this incident, except those configured on the Interface Settings tab.

(*HP Network Node Manager iSPI Network Engineering Toolset Software*) The Diagnostic Selections form enables you to configure NNMi to automatically gather NNM iSPI NET diagnostic information for the Incident you are configuring. When using this form, you specify the diagnostics you want to run on each applicable node in the specified Node Group.

To configure Diagnostics to run on a Source Node for an incident:

1. Navigate to the **Diagnostics Selection** form.
   1. From the workspace navigation panel, select the **Configuration** workspace.
   2. Expand the **Incidents** folder.
   3. Select **Syslog Message Configurations**.
   4. Do one of the following:
      - To create an Incident configuration, click the ✳ New icon.
      - To edit an Incident configuration, select a row, click the 📂 Open icon, and continue.
   5. Navigate to **Node Settings** tab, and do one of the following:
      - To create a Node Settings configuration, click the ✳ New icon.
      - To edit a Node Settings configuration,select a row, click the 📂 Open icon, and continue.
      - To delete a Node Settings configuration, select the Node setting, and click the ✖ Delete icon.
   6. Navigate to the **Diagnostic Selection** tab, and do one of the following:
      - To create a Diagnostic Selection setting, click the ✳ New icon, and continue.
      - To edit a Diagnostic Selection setting, select a row, click the 📂 Open icon, and continue.
      - To delete a Diagnostic Selection setting, select a row, and click the ✖ Delete icon.
2. Provide the required information (see table).
3. Click 💾 **Save and Close** to save your changes and return to the previous form.

After you configure the Diagnostic for the incident and Node Group, the Diagnostic must match the following criteria before the Diagnostic runs:

- The Source Node must be in the specified Node Group.
- The Diagnostic must be valid for the Source Node. (For example, only Nortel switch Diagnostics are run on Nortel switches.)
- The incident's current lifecycle state must match a lifecycle state for which it was configured. (For example, if you configure the Incident to run a specified Diagnostic when the incident is Closed, then if the current Incident's Lifecycle State is Closed, NNMi runs that Diagnostic.)

Note: If a Source Node is in more than one Node Group, the Diagnostic is only run on the node the first time NNMi finds a match for that Source Node based on the configuration Ordering field.

If these criteria are met, NNM iSPI NET runs the Diagnostics and generates Diagnostic reports to help you solve the problem on the Source Node.

After you configure Diagnostics for an incident, you can also run Diagnostics and access the Diagnostics reports on demand, using **Actions** → **Run Diagnostics (iSPI NET only)** in the Incident form. The same criteria apply (see the criteria above). See Incident Form:Diagnostics Tab for more information.

> **Tip:** You can right-click any object in a table or map view to access the **Actions** menu.

You can also run and access Diagnostics reports from a Node form. See Node Form: Diagnostics Tab for more information.

Diagnostic Settings Attributes

Attribute	Description
Flow Definition	Select the Diagnostic (Flow Definition) you want to use for the specified Node Group. Click the Lookup icon and choose one of the following options: • Show Analysis to display Analysis Pane information for the Flow Definition name displayed. (See Use the Analysis Pane for more information about the Analysis Pane.) • Quick Find to view the list of possible diagnostic Flow Definitions. NNMi provides diagnostics for the following types of devices: ▪ Cisco switch ▪ Cisco router ▪ Cisco switch/router ▪ Nortel switch See "Diagnostics (Flows) Provided by NNM iSPI NET" on page 758 for more information about the diagnostics provided and the devices to which they apply.
Lifecycle State	Incident Lifecycle State of the target Incident. If the incident's Lifecycle State matches the value specified here, the Diagnostic runs. The Diagnostic automatically runs on each applicable Source Node in the specified Node Group if the incident has the Lifecycle State currently configured in this attribute of the Diagnostic (Flow Definition - set of automated commands).
Enable	Use this attribute to temporarily disable an incident's Diagnostics settings: **Enable** □ = Temporarily disable the selected configuration. **Enable** ☑ = Enable the selected configuration.

Configure Suppression Settings for a Syslog Message Incident (HP ArcSight)

For information about each Syslog Message tab:

NNMi enables you to suppress incidents based on Interface Group, Node Group, or default Suppression settings. NNMi applies your Suppression settings in the following order. Only the first match applies.

1. Interface Group (Management Event Configuration Form: Interface Settings tab)
2. Node Group (Management Event Configuration Form: Node Settings tab)
3. Suppression configuration settings without specifying an Interface Group or Node Group (Management Event Configuration Form: Suppresion tab)

A Payload Filter enables you to use the data that is included with any of the following items before they are stored as incidents in NNMi:

- Traps generated from an SNMP agent
- Syslog messages generated from `ArcSightEvent` (HP ArcSight only)
- Management incidents that are generated by NNMi

Examples of the type of data that can be used as a Payload Filter include SNMP trap varbind names and values as well as CIA (Custom Incident Attribute) names and values. For example, you might want NNMi to suppress a particular status change notification trap for a specified Node Group or Interface Group. To do so, you could include the name of the trap varbind that stores this information as well as the particular status change value string the traps that you want to suppress should contain.

See "Configure Incident Suppression Settings for an Interface Group (Syslog Message)(HP ArcSight)" on page 955 for information about how to suppress an incident for an Interface Group with or without a Payload Filter.

See "Configure Incident Suppression Settings for a Node Group (Syslog Message) (HP ArcSight)" on page 992 for more information about how to suppress an incident for a Node Group with or without a Payload Filter.

To configure suppression for an incident using a Payload Filter without an Interface Group or Node Group Filter:

1. Navigate to the **Management Event Configuration** form:
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Expand the **Incidents** folder.
   c. Select **Management Event Configurations**.
   d. Do one of the following:
      i. To create an incident configuration, click the ✱ New icon, and continue.
      ii. To edit an incident configuration, select a row, click the 📂 Open icon, and continue.

    iii. To delete an incident configuration, select a row, and click the ✖ Delete icon.

2. Select the **Suppression** tab.
3. Provide the required information (see table)
4. Click **Save and Close** to save your changes and return to the previous form.

Suppression Attributes

Name	Description
Enable	Use this attribute to temporarily disable an incident's suppression settings: **Enable** □ = Temporarily disable the selected configuration. **Enable** ☑ = Enable the selected configuration.

Payload Filter	The Payload Filter Editor enables you to create expressions that further refine the filters used to select the incidents to be suppressed, enriched, or dampened. Make sure to design any complex Payload Filters offline as a Boolean expression first. This method can help to minimize errors when entering your expressions using the Payload Filter editor.
	When creating a Payload Filter, note the following:
	• Payload Filter expressions for the `like` and `not like` operators use the syntax defined for java regular expressions (java.util.regex Pattern class).
	• You must use a `ciaName` that already exists in the trap or event you are configuring.
	• Each set of expressions associated with a Boolean Operator is treated as if it were enclosed in parentheses and evaluated together.
	• View the expression displayed under **Filter String** to see the logic of the expression as it is created.
	• The `AND` and `OR` Boolean Operators must contain at least two expressions as shown in the example below.
	The following example filters incidents on voltage state:
	`AND`
	`    ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7`
	`    ciaValue = 5`
	NNMi evaluates the expression above as follows:
	`(ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7 AND ciaValue = 5)`
	NNMi finds all incidents with a varbind `.1.3.6.1.4.1.9.9.13.1.2.1.7` value of **5**.
	• The placement of your cursor and the subsequent text that is selected is important when performing operations using the Payload Filter Editor. For example, you append to, replace, or change the indentation of the expression that is selected.
	• The placement of your cursor and the subsequent text that is selected is especially

Suppression Attributes , continued

Name	Description

important when adding your Boolean operators.

- You can include more than one varbind in the same Payload Filter expression as shown in the following example:

  ```
  ((ciaName like \Q.1.3.6.1.4.1.9.9\E.* AND ciaValue = 25) AND
  (ciaName like \Q.1.3.6.1.2.1.2.2.1.1.3\E.* AND ciaValue = 3))
  ```

  In this example, a given trap must meet each of the following criteria:

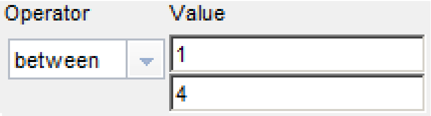
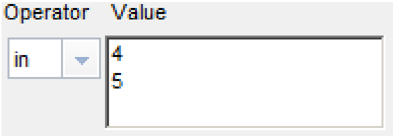
  - Contain a varbind whose Object Identifier (OID) matches the regular expression `\Q.1.3.6.1.4.1.9.9\E.*` and has a value of `25`.
  - Contain a varbind whose OID matches the regular expression `\Q.1.3.6.1.2.1.2.2.1.1.3\E.*` and has a value of `3`.

Payload Filter Editor Components

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: • ciaName • ciaValue
Operator	Valid operators are described below. • **=** Finds all values equal to the value specified. Click here for an example. Example: `ciaName=.1.3.6.1.4.1.9.9.13.1.2.1.7` matches any incident that contains a varbind with the name value of **.1.3.6.1.4.1.9.9.13.1.2.1.7**. • **!=** Finds all values not equal to the value specified. Click here for an example. Example: `ciaName! = .1.3.6.1.4.1.9.9.13.1.2.1.7` matches any incident that contains a varbind with a name value other than **1.3.6.1.4.1.9.9.13.1.2.1.7**. • **<** Finds all values less than the value specified. Click here for an example. Example: `ciaValue < 6` matches any incident that contains a varbind with a value less than **6**. • **<=** Finds all values less than or equal to the value specified. Click here for an example. Example: `ciaValue <= 6` matches any incident that contains a varbind with a value less than or equal to **6**.

Suppression Attributes , continued

Name	Description
	Payload Filter Editor Components, continued

Attribute	Description
	• **>** Finds all values greater than the value specified. Click here for an example. Example: `ciaValue > 4` matches any incident that contains a varbind with a value greater than **4**. • **>=** Finds all values greater than or equal to the value specified. Click here for an example. Example: `ciaValue >= 4` matches any incident that contains a varbind with values greater than or equal to **4**. • **between** Finds all values equal to and between the two values specified. Click here for an example. Example: `ciaValue between`  Operator: between Value: 1 4 matches any incident that contains a varbind value equal to or greater than **1** and equal to or less than **4**. **Note**: As shown in the example, each value must be entered on a separate line. • **in** Finds any match to at least one value in a list of values. Click here for an example. Example: `ciaValue in`  Operator: in Value: 4 5 matches any incident with a varbind value of either **4** or **5**. **Note**: As shown in the example, each value must be entered on a separate line. NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example (**4, 5**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line.

Suppression Attributes , continued

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td></td><td>

Payload Filter Editor Components, continued

<table>
<tr><th>Attribute</th><th>Description</th></tr>
<tr><td></td><td>

- **is not null** Finds all non-blank values. Click here for an example.

  Example: `ciaValue is not null` matches any incident with a varbind that contains a value.

- **is null** Finds all blank values. Click here for an example.

  Example: `ciaValue is null` matches any incident with a varbind that does not contain a value.

- **like** Finds matches using the syntax defined for java regular expressions. See the Pattern (Java Platform SE6) API documentation at : `http://download.oracle.com/javase/6/docs/api/java/util/regex/Pattern.html` for more information. Click here for more information.

  The period asterisk (.*) characters mean *any number of characters of any type at this location*.

  The period (.) character means *any single character of any type at this location*.

  Note: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below.

  Example:

  `ciaName like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that begins with 1.3.6.1.4.1.9.9 and (optionally) ends with any number of characters.

  `ciaValue like .*Chicago.*` finds all traps or events that contain a varbind value that includes the string **Chicago**.

- **not between** Finds all values except those between the two values specified. Click here for an example.

  Example: `ciaValue not between 5 8` matches an incident that contains a varbind with the values less than **5** or greater than **8** .

- **not in** Finds all values except those included in the list of values. Click here for an example.

  Example:

  `ciaValue not in`

</td></tr>
</table>

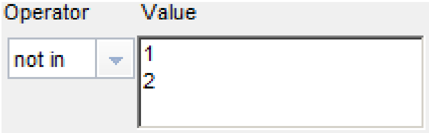
</td></tr>
</table>

Suppression Attributes , continued

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td></td><td>

Payload Filter Editor Components, continued

<table>
<tr><th>Attribute</th><th>Description</th></tr>
<tr><td></td><td>



Operator: not in

Value:
1
2

matches any incident that contains a varbind with values other than **1** and **2**.

Note: As shown in the example, each value must be entered on a separate line.

NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example, (**1, 2**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line.

- **not like** Finds all that do not have the values specified using the syntax defined for Java regular expressions. See the Pattern (Java Platform SE6) API documentation at: `http://download.oracle.com/javase/6/docs/api/java/util/regex/Pattern.html` for more information. Click here for an example.

The period asterisk (.*) characters mean *any number of characters of any type at this location*.

The period (.) character means *any single character of any type at this location*.

Note: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below.

Example:

`ciaName not like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that does not begin with **.1.3.6.1.4.1.9.9** and (optionally) ends with any number of characters.

`ciaValue not like .*Chicago.*` finds all traps or events that do not contain a varbind value that includes the string **Chicago**.

</td></tr>
<tr><td>Value</td><td>

The value for which you want NNMi to search.

Note the following:

- The values you enter are case sensitive.

</td></tr>
</table>

</td></tr>
</table>

Suppression Attributes , continued

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td></td><td>

Payload Filter Editor Components, continued

<table>
<tr><th>Attribute</th><th>Description</th></tr>
<tr><td></td><td>

- NNMi displays a variable number of value fields depending on the Operator selected. For example, the `between` Operator causes two value fields to be displayed.
- The `between, in` and `not in` operators require that each value be entered on a separate line.

</td></tr>
</table>

Payload Filter Editor Buttons

<table>
<tr><th>Button</th><th>Description</th></tr>
<tr><td>Append</td><td>Appends the current expression (Attribute, Operator,and Value) to the selected expression already included in the filter string.</td></tr>
<tr><td>Insert</td><td>Inserts the current expression (Attribute, Operator,and Value) in front of the cursor location within the Filter String.</td></tr>
<tr><td>Replace</td><td>Replaces the selected expression with the expression displayed in the Attribute, Operator, and Value fields.</td></tr>
<tr><td>AND</td><td>Inserts the AND Boolean Operator in the selected cursor location.
<b>Note</b>: View the expression displayed under <b>Filter String</b> to see the logic of the expression as it is created.</td></tr>
<tr><td>OR</td><td>Inserts the OR Boolean Operator in the current cursor location.
<b>Note</b>: View the expression displayed under <b>Filter String</b> to see the logic of the expression as it is created.</td></tr>
<tr><td>NOT</td><td>Can be used in any part of the Filter String to specify that NNMi should exclude interfaces with values that pass the expression that immediately follows the <code>NOT</code>.
For example, when evaluating the following expression, NNMi includes interfaces with (interface description) <code>ifDesc</code> containing <b>VLAN</b>, and excludes any Interfaces that have <b>VLAN10</b> for the (interface name) <code>ifName</code> value:
<code>(ifDesc like VLAN AND NOT (ifName=VLAN10))</code>
<b>Note</b>: View the expression displayed under Filter String to see the logic of the expression as it is created .</td></tr>
<tr><td>EXISTS</td><td>Used for filters that include Capabilities or Custom Attribute names</td></tr>
</table>

</td></tr>
</table>

Suppression Attributes , continued

<table>
<tr><th>Nam
e</th><th>Description</th></tr>
<tr><td></td><td>

Payload Filter Editor Buttons, continued

<table>
<tr><th>Button</th><th>Description</th></tr>
<tr><td></td><td>and values in the Filer String.
Indicates that you want NNMi to consider interfaces that do not have any Capabilities or Custom Attributes when evaluating the Filter String.
<b>Note</b>: If you include Capabilities or Custom Attribute names and values in the Filer String, but do not use <b>EXISTS</b> or <b>NOT EXISTS</b>, NNMi excludes from its search interfaces that do not include Capabilities or Custom Attributes.
For example, when evaluating the following Filter String, NNMi includes interfaces with (interface description) <code>ifDesc</code> containing <b>VLAN</b>, as well as any Interfaces Custom Attribute Role value is <b>LAN Connection to Oracle Server</b>:
<code>(ifDesc like VLAN OR EXISTS((customAttrName=Role AND customAttrValue=LAN Connection to Oracle Server)))</code>
<b>Note</b>: View the expression displayed under Filter String to see the logic of the expression as it is created.</td></tr>
<tr><td>NOT EXISTS</td><td>Used for filters that include Capabilities or Custom Attribute names and values in the Filer String. Indicates that you want NNMi to consider interfaces that do not have any Capabilities or Custom Attributes when evaluating the Filter String, but exclude the interfaces that match the expression that follows the <b>NOT EXISTS</b>.
<b>Note</b>: If you include Capabilities or Custom Attribute names and values in the Filer String, but do not use <b>EXISTS</b> or <b>NOT EXISTS</b>, NNMi excludes from its search interfaces that do not include Capabilities or Custom Attributes.
For example, when evaluating the following expression, NNMi includes interfaces with (interface description) <code>ifDesc</code> containing <b>VLAN</b>, and excludes any Interfaces that have the Custom Attribute Role and that Role value is <b>LAN Connection to Oracle Server</b>:
<code>(ifDesc like VLAN OR NOT EXISTS ((customAttrName=Role AND customAttrValue=LAN Connection to Oracle Server)))</code>
<b>Note</b>: View the expression displayed under Filter String to see the logic of the expression as it is created.</td></tr>
</table>

</td></tr>
</table>

Suppression Attributes , continued

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td></td><td>
<b>Payload Filter Editor Buttons, continued</b>
<table>
<tr><th>Button</th><th>Description</th></tr>
<tr><td>Delete</td><td>Deletes the selected expression.
<b>Note</b>: If the Boolean Operator is selected, the Payload Filter Editor deletes all expressions associated with the Boolean Operator.</td></tr>
</table>
</td></tr>
</table>

Configure Enrichment Settings for a Syslog Message Incident (HP ArcSight)

For information about each Syslog Message tab:

NNMi enables you to fine tune and enhance incidents based on Interface Group, Node Group, or default Enrichment settings. NNMi applies your Enrichment settings in the following order. Only the first match applies.

1. Interface Group (Management Event Configuration Form: Interface Settings tab)
2. Node Group (Management Event Configuration Form: Node Settings tab)
3. Enrich configuration settings without specifying an Interface Group or Node Group (Management Event Configuration Form: Enrichment tab)

The types of items you can fine tune and enhance for a selected incident configuration, include:

- Category
- Family
- Severity
- Priority
- Correlation Nature
- Message
- Assigned To

Note: Any configuration you specify for Severity, Priority, or Message overrides those values provided in the Management Event Configuration Form: Basics information.

A Payload Filter enables you to use the data that is included with any of the following items before they are stored as incidents in NNMi:

- Traps generated from an SNMP agent
- Syslog messages generated from `ArcSightEvent` (HP ArcSight only)

- Management incidents that are generated by NNMi
- Events generated by NNM 6.x or 7.x management stations

Examples of the type of data that can be used as a Payload Filter include SNMP trap varbind names and values as well as CIA (Custom Incident Attribute) names and values. For example, you might want NNMi to suppress a particular status change notification trap for a specified Node Group or Interface Group. To do so, you could include the name of the trap varbind that stores this information as well as the particular status change value string the traps that you want to suppress should contain.

Note: The CIA added to an incident must be provided by NNMi. You cannot create CIAs.

See "Configure Incident Enrichment Settings for an Interface Group (Syslog Message)(HP ArcSight)" on page 963 for information about how to enrich an incident for an Interface Group with or without a Payload Filter.

See "Configure Incident Enrichment Settings for a Node Group (Syslog Message) (HP ArcSight)" on page 1000 for more information about how to enrich an incident for a Node Group with or without a Payload Filter.

To configure Enrichment settings for an incident using a Payload Filter without an Interface Group or Node Group Filter:

1. Navigate to the **Management Event Configuration** form:
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Expand the **Incidents** folder.
   c. Select **Management Event Configurations** .
   d. Do one of the following:
      i. To create an incident configuration, click the ✳ New icon, and continue.
      ii. To edit an incident configuration, select a row, click the 📂 Open icon, and continue.
      iii. To delete an incident configuration, select a row, and click the ✖ Delete icon.
2. Select the **Enrichment** tab.
3. Do one of the following:
   a. To create a new configuration, click the ✳ New icon.
   b. To edit an existing configuration, select a row, click the 📂 Open icon, and continue.
4. Provide the required information (see table)
5. Click 💾 **Save and Close** to save your changes and return to the previous form.

Enrichment Attributes

Name	Description
Category	Use the Category attribute to customize the category for this incident configuration. Possible values include: • Accounting

Enrichment Attributes , continued

Name	Description
	- Application Status - Configuration - Fault - Performance - Security - Status See "Specify Category and Family Attribute Values for Organizing Your Incidents (Syslog Message) (HP ArcSight)" on page 941 for more information.
Family	Use the Family attribute to customize the Family for this incident configuration. Select from the drop-down list or create a new value. For example, some of the values provided by NNMi include: - Address - Aggregated Port (Interfaces using **Link Aggregation**[1] protocol. See Interface Form: Link Aggregation tab.) - Card - Connection - Correlation - Interface - Node See "Specify Category and Family Attribute Values for Organizing Your Incidents (Syslog Message) (HP ArcSight)" on page 941 for more information.
Severity	The incident Severity represents the seriousness calculated for the incident. Use the Severity attribute to specify the severity that should be assigned to the incident you are configuring. Possible values are described below: **Normal** - Indicates there are no known problems related to the associated object. This Severity is meant to be informational. Generally, no action is needed for these incidents. **Warning** - Indicates there might be a problem related to the associated object.

[1] Protocols used on Switches to configure multiple Interfaces (Aggregation Member Interfaces) to function as if they were one (an Aggregator Interface). When two Aggregator Interfaces establish a connection, that connection is an Aggregator Layer 2 Connection. The Aggregator Layer 2 Connection appears on Layer 2 Neighbor View maps as a thick line with an Interface icon at each end (representing the Aggregator Interface). When you double-click the thick line, it converts into multiple thin lines representing the participating Aggregation Member Layer 2 Connections with their Aggregation Member Interfaces at each end of the lines.

Enrichment Attributes , continued

Name	Description
	Minor - Indicates NNMi has detected problems related to the associated object that require further investigation. **Major** - Indicates NNMi has detected problems related to the associated object to be resolved before they become critical. **Critical** - Indicates NNMi has detected problems related to the associated object that require immediate attention.
Priority	Used to communicate the urgency of resolving the selected incident. You control this value. NNMi sets this value to null by default. The lower the number the higher the priority. Possible values are: 5 **None** 4 **Low** 3 **Medium** 2 **High** 1 **Top** **Note**: The icons are displayed only in table views.
Correlation Nature	Use the Correlation Nature to customize the Correlation Nature for this incident configuration. Possible values include: • Info • None • Root Cause • Secondary Root Cause • Symptom • Stream Correlation • Service Impact • Dedup Stream Correlation • Rate Stream Correlation See Incident Form: General Tab for more information.
Message Format	When configuring an incident, specify how the incident message appears in the incident view. The string you specify in the Message Format attribute is visible in an incident view. **Note**: The incident message limit is 1024 characters. If you exceed this limit, NNMi truncates the value starting from the right.

Enrichment Attributes , continued

Name	Description
	You can use any combination of default and custom attributes: "Valid Parameters for Configuring Incident Messages (Syslog Message) (HP ArcSight)" on page 946 "Include Custom Incident Attributes in Your Message Format (Syslog Message) (HP ArcSight)" on page 953
Assigned To	Use to specify the owner of any incident generated for this incident configuration. Click the Lookup icon and select Quick Find to select a valid user name. **Note**: You can also begin to type a valid user name and use the auto-complete feature to select the user name of interest.
Description	Use the Description attribute to provide additional information that you want to note about the current enhancement configuration. This description applies only to the enhancement configuration and does not appear when NNMi displays any associated incident. Type a maximum of 1024 characters. Alpha-numeric and special characters (~ ! @ # $ % ^ & * ( ) _+ -) are permitted.

Configure Dampening Settings for a Syslog Message Incident (HP ArcSight)

For information about each Syslog Message tab:

NNMi enables you to delay the following for an incident configuration based on the Source Object's participation in an Interface Group:

- Execution of Incident Actions
- Execution of Diagnostics (*HP Network Node Manager iSPI Network Engineering Toolset Software \ NNM iSPI NET*)
- Appearance within Incident views in the NNMi Console

You can configure the Dampening settings based on Interface Group, Node Group, or default Dampening settings. NNMi applies your Dampening settings in the following order. Only the first match applies.

1. Interface Group (Management Event Configuration Form: Interface Settings tab)
2. Node Group (Management Event Configuration Form: Node Settings tab)
3. Dampening configuration settings without specifying an Interface Group or Node Group (Management Event Configuration Form: Dampening tab)

When using the Dampening configuration, note the following:

- Duplicate and Rate Correlation incidents inherit the Dampening settings from their Correlated Children. If the Correlated Children are Closed while Dampened, and therefore deleted, NNMi retains the parent Duplicate or Rate Correlation incident. See "Correlate Duplicate Incidents (Deduplication Configuration)" on page 659 and "Track Incident Frequency (Rate: Time Period and Count)" on page 659 for more information about Duplicate and Rate Correlation incidents.

  Note: NNMi administrators can view the number of incidents Closed and deleted while dampened. Access the **Help** → **System Information** → **Health** tab, click the View Detailed Health Report button, and search for the word dampened.

- After the Dampen Interval has passed, NNMi changes the Lifecycle State to REGISTERED.

  See About the Incident Lifecycle for more information about Lifecycle State.

See "Configure Incident Dampening Settings for an Interface Group (Syslog Message) (HP ArcSight)" on page 975 for information about how to configure Dampening settings for an Interface Group with or without a Payload Filter.

See "Configure Incident Dampening Settings for a Node Group (Syslog Message) (HP ArcSight)" on page 1012 for more information about how to configure Dampening settings for a Node Group with or without a Payload Filter.

To configure Dampening settings for an incident using a Payload Filter without an Interface Group or Node Group Filter:

1. Navigate to the **Syslog Message Configuration** form:
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Expand the **Incidents** folder.
   c. Select **Syslog Message Configurations**.
   d. Do one of the following:
      i. To create a configuration, click the ✱ New icon, and continue.
      ii. To edit configuration, double-click the row representing the configuration you want to edit, and continue.
      iii. To delete a configuration, select a row, and click the ✖ Delete icon.
2. Select the **Dampening** tab.
3. Provide the required information (see table)
4. Click 💾 **Save and Close** to save your changes and return to the previous form.

Dampening Attributes

Name	Description
Enable	Use this attribute to temporarily disable an incident's Dampening settings: **Enable** □ = Temporarily disable the selected configuration. **Enable** ☑ = Enable the selected configuration.

Dampening Attributes , continued

Name	Description
Hour	Specifies the number of hours to be used for the Dampen Interval.
Minutes	Specifies the number of minutes to be used for the Dampen Interval.
Seconds	Specifies the number of seconds to be used for the Dampen Interval.

Payload Filter	The Payload Filter Editor enables you to create expressions that further refine the filters used to select the incidents to be suppressed, enriched, or dampened. Make sure to design any complex Payload Filters offline as a Boolean expression first. This method can help to minimize errors when entering your expressions using the Payload Filter editor.
	When creating a Payload Filter, note the following:
	▪ Payload Filter expressions for the `like` and `not like` operators use the syntax defined for java regular expressions (java.util.regex Pattern class).
	▪ You must use a `ciaName` that already exists in the trap or event you are configuring.
	▪ Each set of expressions associated with a Boolean Operator is treated as if it were enclosed in parentheses and evaluated together.
	▪ View the expression displayed under **Filter String** to see the logic of the expression as it is created.
	▪ The `AND` and `OR` Boolean Operators must contain at least two expressions as shown in the example below.
	The following example filters incidents on voltage state:
	`AND`
	`    ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7`
	`    ciaValue = 5`
	NNMi evaluates the expression above as follows:
	`(ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7 AND ciaValue = 5)`
	NNMi finds all incidents with a varbind `.1.3.6.1.4.1.9.9.13.1.2.1.7` value of **5**.
	▪ The placement of your cursor and the subsequent text that is selected is important when performing operations using the Payload Filter Editor. For example, you append to, replace, or change the indentation of the expression that is selected.
	▪ The placement of your cursor and the subsequent text that is selected is especially important when adding your Boolean operators.
	▪ You can include more than one varbind in the same Payload Filter expression as shown in the following example:

Dampening Attributes , continued

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td></td><td>
<code>((ciaName like \Q.1.3.6.1.4.1.9.9\E.* AND ciaValue = 25) AND (ciaName like \Q.1.3.6.1.2.1.2.2.1.1.3\E.* AND ciaValue = 3))</code>

In this example, a given trap must meet each of the following criteria:

○ Contain a varbind whose Object Identifier (OID) matches the regular expression <code>\Q.1.3.6.1.4.1.9.9\E.*</code> and has a value of <code>25</code>.

○ Contain a varbind whose OID matches the regular expression <code>\Q.1.3.6.1.2.1.2.2.1.1.3\E.*</code> and has a value of <code>3</code>.

<b>Payload Filter Editor Components</b>
<table>
<tr><th>Attribute</th><th>Description</th></tr>
<tr><td>Attribute</td><td>The attribute name on which NNMi searches. Filterable attributes include the following:
▪ ciaName
▪ ciaValue</td></tr>
<tr><td>Operator</td><td>Valid operators are described below.

▪ <b>=</b> Finds all values equal to the value specified. Click here for an example.

Example: <code>ciaName=.1.3.6.1.4.1.9.9.13.1.2.1.7</code> matches any incident that contains a varbind with the name value of <b>.1.3.6.1.4.1.9.9.13.1.2.1.7</b>.

▪ <b>!=</b> Finds all values not equal to the value specified. Click here for an example.

Example: <code>ciaName! = .1.3.6.1.4.1.9.9.13.1.2.1.7</code> matches any incident that contains a varbind with a name value other than <b>1.3.6.1.4.1.9.9.13.1.2.1.7</b>.

▪ <b>&lt;</b> Finds all values less than the value specified. Click here for an example.

Example: <code>ciaValue &lt; 6</code> matches any incident that contains a varbind with a value less than <b>6</b>.

▪ <b>&lt;=</b> Finds all values less than or equal to the value specified. Click here for an example.

Example: <code>ciaValue &lt;= 6</code> matches any incident that contains a varbind with a value less than or equal to <b>6</b>.</td></tr>
</table>
</td></tr>
</table>

Dampening Attributes , continued

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td></td><td>

Payload Filter Editor Components, continued

<table>
<tr><th>Attribute</th><th>Description</th></tr>
<tr><td></td><td>

- **>** Finds all values greater than the value specified. Click here for an example.

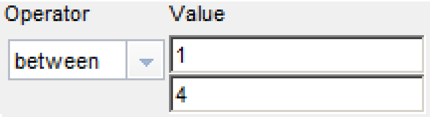
  Example: `ciaValue > 4` matches any incident that contains a varbind with a value greater than **4**.

- **>=** Finds all values greater than or equal to the value specified. Click here for an example.

  Example: `ciaValue >= 4` matches any incident that contains a varbind with values greater than or equal to **4**.

- **between** Finds all values equal to and between the two values specified. Click here for an example.

  Example: `ciaValue between`

  Operator: between
  Value: 1
  4

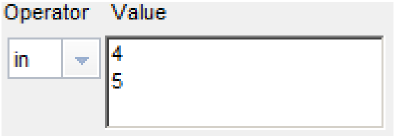
  matches any incident that contains a varbind value equal to or greater than **1** and equal to or less than **4**.

  Note: As shown in the example, each value must be entered on a separate line.

- **in** Finds any match to at least one value in a list of values. Click here for an example.

  Example:

  `ciaValue in`

  Operator: in
  Value: 4
  5

  matches any incident with a varbind value of either **4** or **5**.

  Note: As shown in the example, each value must be entered on a separate line.

  NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example (**4, 5**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line.

</td></tr>
</table>

</td></tr>
</table>

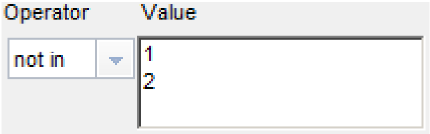
Dampening Attributes , continued

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td></td><td>
<b>Payload Filter Editor Components, continued</b>
<table>
<tr><th>Attribute</th><th>Description</th></tr>
<tr><td></td><td>
<ul>
<li><b>is not null</b> Finds all non-blank values. Click here for an example.
Example: <code>ciaValue is not null</code> matches any incident with a varbind that contains a value.</li>
<li><b>is null</b> Finds all blank values. Click here for an example.
Example: <code>ciaValue is null</code> matches any incident with a varbind that does not contain a value.</li>
<li><b>like</b> Finds matches using the syntax defined for java regular expressions. See the Pattern (Java Platform SE6) API documentation at : <code>http://download.oracle.com/javase/6/docs/api/java/util/regex/Pattern.html</code> for more information. Click here for more information.
The period asterisk (.*) characters mean <i>any number of characters of any type at this location</i>.
The period (.) character means <i>any single character of any type at this location</i>.
<b>Note</b>: To include literal string values in the Value attribute, enclose the string value in \Q&lt;<i>literal_value</i>&gt;\E as shown in the Examples listed below.
Example:
<code>ciaName like \Q.1.3.6.1.4.1.9.9\E.*</code> matches any incident that contains a varbind name value that begins with 1.3.6.1.4.1.9.9 and (optionally) ends with any number of characters.
<code>ciaValue like .*Chicago.*</code> finds all traps or events that contain a varbind value that includes the string <b>Chicago</b>.</li>
<li><b>not between</b> Finds all values except those between the two values specified. Click here for an example.
Example: <code>ciaValue not between 5 8</code> matches an incident that contains a varbind with the values less than <b>5</b> or greater than <b>8</b> .</li>
<li><b>not in</b> Finds all values except those included in the list of values. Click here for an example.
Example:
<code>ciaValue not in</code></li>
</ul>
</td></tr>
</table>
</td></tr>
</table>

Dampening Attributes , continued

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td></td><td>

Payload Filter Editor Components, continued

<table>
<tr><th>Attribute</th><th>Description</th></tr>
<tr><td></td><td>

Operator Value
not in
1
2
matches any incident that contains a varbind with values other than **1** and **2**.
Note: As shown in the example, each value must be entered on a separate line.
NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example, (**1, 2**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line.
■ **not like** Finds all that do not have the values specified using the syntax defined for Java regular expressions. See the Pattern (Java Platform SE6) API documentation at: `http://download.oracle.com/javase/6/docs/api/java/util/regex/Pattern.html` for more information. Click here for an example.
The period asterisk (.*) characters mean *any number of characters of any type at this location*.
The period (.) character means *any single character of any type at this location*.
Note: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below.
Example:
`ciaName not like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that does not begin with **.1.3.6.1.4.1.9.9** and (optionally) ends with any number of characters.
`ciaValue not like .*Chicago.*` finds all traps or events that do not contain a varbind value that includes the string **Chicago**.
</td></tr>
<tr><td>Value</td><td>The value for which you want NNMi to search.
Note the following:</td></tr>
</table>

</td></tr>
</table>

Dampening Attributes , continued

Name	Description

Payload Filter Editor Components, continued

Attribute	Description
	- The values you enter are case sensitive. - NNMi displays a variable number of value fields depending on the Operator selected. For example, the `between` Operator causes two value fields to be displayed. - The `between, in` and `not in` operators require that each value be entered on a separate line.

Payload Filter Editor Buttons

Button	Description
Append	Appends the current expression (Attribute, Operator,and Value) to the selected expression already included in the filter string.
Insert	Inserts the current expression (Attribute, Operator,and Value) in front of the cursor location within the Filter String.
Replace	Replaces the selected expression with the expression displayed in the Attribute, Operator, and Value fields.
AND	Inserts the AND Boolean Operator in the selected cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
OR	Inserts the OR Boolean Operator in the current cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
NOT	Can be used in any part of the Filter String to specify that NNMi should exclude interfaces with values that pass the expression that immediately follows the `NOT`. For example, when evaluating the following expression, NNMi includes interfaces with (interface description) `ifDesc` containing **VLAN**, and excludes any Interfaces that have **VLAN10** for the (interface name) `ifName` value: `(ifDesc like VLAN AND NOT (ifName=VLAN10))` **Note**: View the expression displayed under Filter String to see the logic of the expression as it is created .

Dampening Attributes , continued

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td></td><td>

Payload Filter Editor Buttons, continued

<table>
<tr><th>Button</th><th>Description</th></tr>
<tr><td>EXISTS</td><td>Used for filters that include Capabilities or Custom Attribute names and values in the Filer String.
Indicates that you want NNMi to consider interfaces that do not have any Capabilities or Custom Attributes when evaluating the Filter String.
<b>Note</b>: If you include Capabilities or Custom Attribute names and values in the Filer String, but do not use <b>EXISTS</b> or <b>NOT EXISTS</b>, NNMi excludes from its search interfaces that do not include Capabilities or Custom Attributes.
For example, when evaluating the following Filter String, NNMi includes interfaces with (interface description) <code>ifDesc</code> containing <b>VLAN</b>, as well as any Interfaces Custom Attribute Role value is <b>LAN Connection to Oracle Server</b>:
<code>(ifDesc like VLAN OR EXISTS ((customAttrName=Role AND customAttrValue=LAN Connection to Oracle Server)))</code>
<b>Note</b>: View the expression displayed under Filter String to see the logic of the expression as it is created.</td></tr>
<tr><td>NOT EXISTS</td><td>Used for filters that include Capabilities or Custom Attribute names and values in the Filer String. Indicates that you want NNMi to consider interfaces that do not have any Capabilities or Custom Attributes when evaluating the Filter String, but exclude the interfaces that match the expression that follows the <b>NOT EXISTS</b>.
<b>Note</b>: If you include Capabilities or Custom Attribute names and values in the Filer String, but do not use <b>EXISTS</b> or <b>NOT EXISTS</b>, NNMi excludes from its search interfaces that do not include Capabilities or Custom Attributes.
For example, when evaluating the following expression, NNMi includes interfaces with (interface description) <code>ifDesc</code> containing <b>VLAN</b>, and excludes any Interfaces that have the Custom Attribute Role and that Role value is <b>LAN Connection to Oracle Server</b>:
<code>(ifDesc like VLAN OR NOT EXISTS ((customAttrName=Role AND customAttrValue=LAN Connection to Oracle Server)))</code></td></tr>
</table>

</td></tr>
</table>

Dampening Attributes , continued

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td></td><td>

Payload Filter Editor Buttons, continued

<table>
<tr><th>Button</th><th>Description</th></tr>
<tr><td></td><td><b>Note</b>: View the expression displayed under Filter String to see the logic of the expression as it is created.</td></tr>
<tr><td>Delete</td><td>Deletes the selected expression.
<b>Note</b>: If the Boolean Operator is selected, the Payload Filter Editor deletes all expressions associated with the Boolean Operator.</td></tr>
</table>

</td></tr>
</table>

Configure Deduplication for a Syslog Message Incident (HP ArcSight)

For information about each Syslog Message tab:

The deduplication configuration determines what values NNMi should match to detect when an SNMP trap, Syslog Message (HP ArcSightonly), Management Event, or Remote NNM 6.x/7.x event is a duplicate.

Note the following:

- Suppression, Enrichment, and Dampening are not supported for Deduplication incidents.
- NNMi applies only one deduplication configuration per incident . If NNMi generates an incident using a specified deduplication configuration, NNMi continues to correlate duplicate incidents using the original configuration. To use a different deduplication configuration for an incident, first delete the current deduplication incident (created using the original deduplication configuration). NNMi generates the next deduplication incident according to the new deduplication configuration settings.
- By default, NNMi updates the **Duplicate Count** every 30 seconds. This interval cannot be changed.
- NNMi continues to update the duplicate count regardless of an incident's lifecycle state. For example, if an incident's **Lifecycle State** is set to **Closed**, the duplicate count continues to be incremented. See About the Incident Lifecycle for more information. This behavior helps you identify situations in which the incident is not yet fixed. Take note if the Duplicate Count is incremented after a lengthy time period has elapsed, which might indicate there is a new problem with the node, interface, or address.
- Each time you stop and restart ovjboss, any incidents that have not yet been correlated or persisted are lost. This means that after a restart of ovjboss, an incoming incident might not be correlated as expected. For example, after a restart of ovjboss, a duplicate incident might not be correlated under its original parent incident. Instead, a new parent incident might be generated. See "Stop or Start an NNMi Process" on page 82for more information about starting and stopping the ovjboss process.

- If a Duplicate Correlation Incident is dampened, note the following:
  - Duplicate Correlation Incidents inherit the Dampening settings from its Correlated Children.
  - NNMi always retains the Parent Duplicate Correlation incident, even if its Child Incidents are Closed and subsequently deleted.

    See "Dampening Incident Configurations" on page 679 for more information about Dampening an incident configuration.

To specify or delete a deduplication configuration:

1. Navigate to the **Syslog Message Configuration** form:
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Expand the **Incidents** folder.
   c. Select **Syslog Message Configurations**.
   d. Do one of the following:
      i. To create a deduplication configuration, click the New icon, and continue.
      ii. To edit a deduplication configuration, select a row, click the Open icon, and continue.
      iii. To delete a deduplication configuration, select a row, and click the Delete icon.
2. Select the **Deduplication** tab.
3. Provide the required information (see "Deduplication Attributes" table).
4. Click **Save and Close** to save your changes and return to the previous form.

Deduplication Attributes

Name	Description
Enabled	Use this attribute to temporarily disable an incident's deduplication configuration: **Enable** □ = Temporarily disable the selected configuration. **Enable** ☑ = Enable the selected configuration. **Note:** After a deduplication configuration is enabled, NNMi increments the **Duplicate Count** for an associated incident regardless of the **Lifecycle State** value. For example, if an incident's **Lifecycle State** is set to **Closed**, the duplicate count continues to be incremented. See About the Incident Lifecycle for more information.
Count	Specifies the number of duplicate incidents for the current configuration that NNMi stores at one time. For example, if the Count is 10, after NNMi receives 10 duplicate incidents, NNMi deletes the first (oldest) duplicate incident and keeps the eleventh. (NNMi stores ten maximum.)
Hours	Used with the Minute and Second Intervals to specify the time that must elapse before a new duplicate incident is generated for this incident configuration. For

Deduplication Attributes, continued

Name	Description
	example, if the Hour Interval value is 1, and no Minute or Second Intervals are specified, and the duplicate incident is not generated within one hour, NNMi generates a new duplicate incident the next time it occurs.
Minutes	Used with the Hour and Second interval to specify the time that must elapse before a new duplicate incident is generated for this incident configuration. For example, if the Minute Interval is 30 and no Hour or Second Intervals are specified, and the duplicate incident is not generated within 30 minutes, NNMi generates a new duplicate incident the next time it occurs.
Seconds	Used with the Hour and Minute Intervals to specify the time that must elapse before a new duplicate incident is generated for this incident configuration. For example, if the Second Interval is 120 and no Hour or Minute Intervals are specified, and the duplicate incident is not generated within 120 seconds, NNMi generates a new duplicate incident the next time it occurs.
Parent Incident	varUsed to specify the Incident Configuration that will be the Parent Incident for the incident you are configuring. For example, you might have created a Management Event Incident Configuration that could be used as the **Parent Incident** for SNMP Trap Incidents. When specifying the **Parent Incident**, you have the following options: • When you want to use a configuration that NNMi provides, use the default **Duplicate Correlation** incident configuration . If you select this option, the incident message for the Parent Incident begins as follows: `Duplicate Correlation for` *incident_configuration_name*> For example if you are configuring a **Node Down** incident and select **Duplicate Correlation** as the **Parent Incident**, the Parent Incident message begins with: **Duplicate Correlation for Node Down**. Each **Node Down** incident that is a duplicate then appears correlated under the **Duplicate Correlation for Node Down** incident. • NNMi also enables you to customize the Parent Incident for a given deduplication scenario. If you have created a Management Event Incident Configuration to use for this deduplication scenario, select the Management Event Incident Configuration that you have created.
Comparison Criteria	Specify the attribute values that must match before the incident is identified as a duplicate. The possible attributes consist of the following choices. • **Name** - The **Name** attribute value from the Incident form: General tab. • **CIA** - Represents any of the following items configured as a Parameter Value using the "Deduplication Comparison Parameters Form " on page 659: ■ The **Value** attribute from the Incident form: Custom Attributes tab ■ An SNMP varbind Object ID ■ An SNMP varbind position number

Deduplication Attributes, continued

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td></td><td>
<p>If you want to use CIA as part of your comparison criteria, see "Deduplication Comparison Parameters Form " on page 659</p>
<ul>
<li><b>SourceNode</b> - The <b>Source Node</b> attribute value from the Basics attributes listed on the Incident form. The Source Node value is the IP Address or Name of the node for which the incident was generated.<p><b>Note</b>: The Source Node must be stored in the NNMi database.</p></li>
<li><b>Source Object</b> - The <b>Source Object</b> attribute value from the Basics attributes listed on the Incident form.<p><b>Note</b>: The Source Object must be stored in the NNMi database.</p></li>
</ul>
<p><b>Note</b>: Each attribute value in the option you select must match before the incident is identified as a duplicate. For example, if you select <b>Name</b>, only the Incident Name value must match. If you select <b>Name SourceNode SourceObject CIA</b>, the Incident Name, Source Node, Source Object, and all Custom Incident Attribute values that you configure as a Parameter Value must match before NNMi identifies the incident as a duplicate.</p>
<p>Selecting an option that includes CIA enables you to further refine the deduplication criteria. For example, you might want to configure deduplication for incidents with CIA values that specify the same State attribute value for a particular network object.</p>
<p>For a description of each Comparison Criteria option, click here.</p>
<table>
<tr><th>Comparison Criteria</th><th>Description</th></tr>
<tr><td>Name</td><td>Value of the <b>Name</b> attribute from the Incident form: General tab must match.</td></tr>
<tr><td>Name CIA</td><td>
<p>Each of the following values must match:</p>
<ul>
<li><b>Name</b> attribute from the Incident form: General tab</li>
<li><b>CIA</b> - Represents the Value associated with any of the following items configured as a Parameter Value using the "Deduplication Comparison Parameters Form " on page 659:
<ul>
<li>Name of a Custom Incident Attribute (CIA) provided by NNMi. (See the Incident form: Custom Attributes tab</li>
<li>An SNMP varbind Object ID</li>
<li>An SNMP varbind position number<p>If you want to use CIA as part of your comparison criteria, see "Deduplication Comparison Parameters Form " on page 659</p></li>
</ul></li>
</ul>
</td></tr>
</table>
</td></tr>
</table>

Deduplication Attributes, continued

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td></td><td>
<table>
<tr><th>Comparison Criteria</th><th>Description</th></tr>
<tr><td>Name SourceNode</td><td><b>Note</b>: Select this option only if the Source Node is stored in the NNMi database.
Each of the following values must match:
• <b>Name</b> attribute from the Incident form: General tab
• The <b>Source Node</b> attribute value from the Basics attributes listed on the Incident form</td></tr>
<tr><td>Name SourceNode CIA</td><td><b>Note</b>: Select this option only if the Source Node is stored in the NNMi database.
Each of the following values must match:
• <b>Name</b> attribute from the Incident form: General tab
• The <b>Source Node</b> attribute value from the Basics attributes listed on the Incident form
• <b>CIA</b> - Represents the Value associated with any of the following items configured as a Parameter Value using the "Deduplication Comparison Parameters Form " on page 659:
▪ The <b>Value</b> attribute from the Incident form: Custom Attributes tab
▪ An SNMP varbind Object ID
▪ An SNMP varbind position number
If you want to use CIA as part of your comparison criteria, see "Deduplication Comparison Parameters Form " on page 659</td></tr>
<tr><td>Name SourceObject</td><td><b>Note</b>: Select this option only if the Source Object is stored in the NNMi database.
Each of the following values must match:
• <b>Name</b> attribute from the Incident form: General tab
• The <b>Source Object</b> attribute value from the Basics attributes listed on the Incident form.</td></tr>
<tr><td>Name SourceObject CIA</td><td><b>Note</b>: Select this option only if the Source Object is stored in the NNMi database.
Each of the following values must match:</td></tr>
</table>
</td></tr>
</table>

Deduplication Attributes, continued

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td></td><td>
<table>
<tr><th>Comparison Criteria</th><th>Description</th></tr>
<tr><td></td><td>• **Name** attribute from the Incident form: General tab
• The **Source Object** attribute value from the Basics attributes listed on the Incident form
• **CIA** - Represents the Value associated with any of the following items configured as a Parameter Value using the "Deduplication Comparison Parameters Form " on page 659:
▪ The **Name** attribute from the Incident form: Custom Attributes tab
▪ An SNMP varbind Object ID
▪ An SNMP varbind position number
If you want to use CIA as part of your comparison criteria, see "Deduplication Comparison Parameters Form " on page 659</td></tr>
<tr><td>Name
SourceNode
SourceObject</td><td>**Note**: Select this option only if the Source Node and Source Object are stored in the NNMi database.
Each of the following values must match:
• **Name** attribute from the Incident form: General tab
• The **Source Node** attribute value from the Basics attributes listed on the Incident form
• The **Source Object** attribute value from the Basics attributes listed on the Incident form</td></tr>
<tr><td>Name
SourceNode
SourceObject
CIA</td><td>**Note**: Select this option only if the Source Node and Source Object are stored in the NNMi database.
Each of the following values must match:
• **Name** attribute from the Incident form: General tab
• The **Source Node** attribute value from the Basics attributes listed on the Incident form
• The **Source Object** attribute value from the Basics attributes listed on the Incident form
• **CIA** - Represents the Value associated with any of the following items configured as a Parameter Value using the "Deduplication Comparison Parameters Form " on page</td></tr>
</table>
</td></tr>
</table>

Deduplication Attributes, continued

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td></td><td>
<table>
<tr><th>Comparison Criteria</th><th>Description</th></tr>
<tr><td></td><td>659:
▪ The <b>Name</b> attribute from the Incident form: Custom Attributes tab
▪ An SNMP varbind Object ID
▪ An SNMP varbind position number
If you want to use CIA as part of your comparison criteria, see "Deduplication Comparison Parameters Form " on page 659</td></tr>
</table>
</td></tr>
<tr><td>Deduplication Comparison Parameters</td><td><i>Optional</i>. If you selected a Comparison Criteria that includes CIA, you must populate one or more rows in this table. See "Deduplication Comparison Parameters Form " on page 659.</td></tr>
</table>

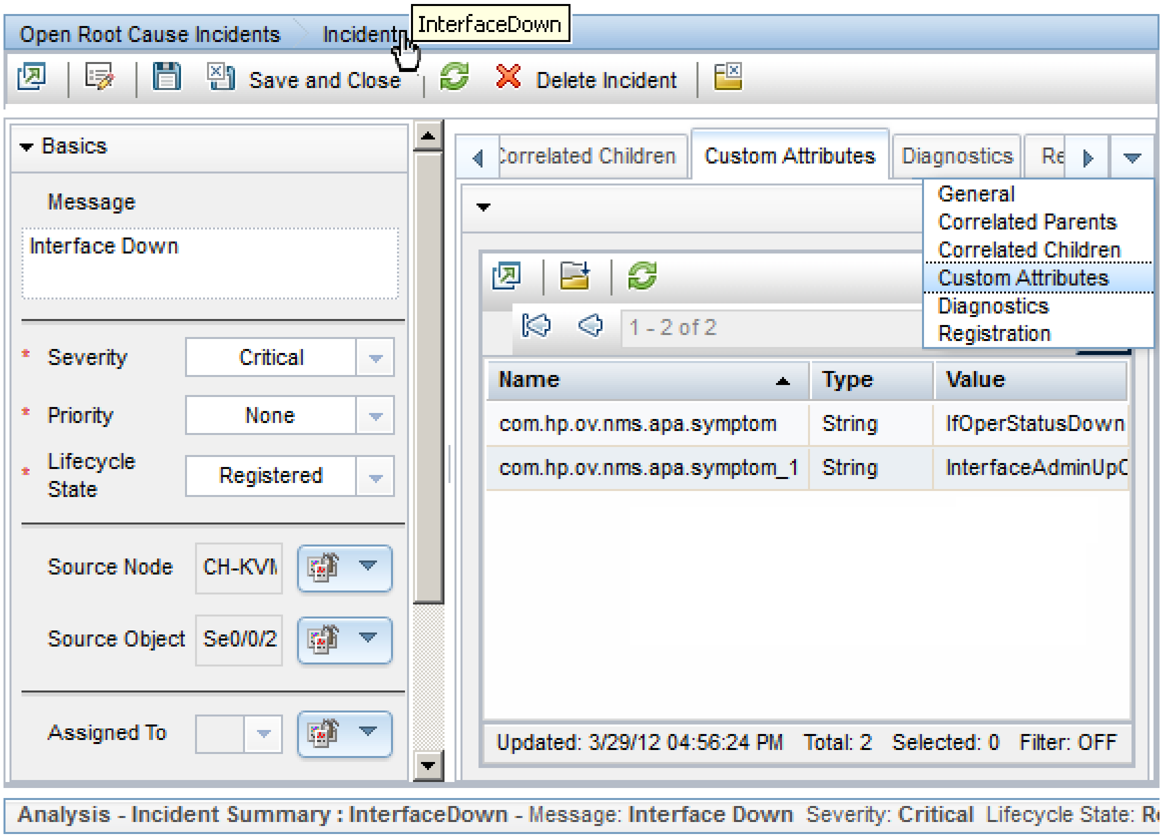
Deduplication Comparison Parameters Form (Syslog Message) (HP ArcSight)

Comparison Parameter values enable accurate identification of duplicate incidents. Custom Incident Attributes (CIAs) are used as Comparison Parameter values. There are two categories of CIAs:

- SNMP trap varbind values (Name = the MIB varbind identifier, Type = asn_*)
- Custom attributes provided by NNMi (Name = cia.*, Type=String). See "Custom Incident Attributes Provided by NNMi (Information for Administrators)" on page 647.

The group of available CIAs depends on which incident you are configuring for this Deduplication (for example, CiscoLinkDown). To see which CIAs are available, navigate to an Incident view, select an instance of that incident-type, click the Open icon and navigate to the Custom Attributes tab. The items listed in the table are the CIAs for that particular incident-type. For example, all CiscoLinkDown incidents would have the same group of CIAs shown in the illustration below.

Note: You can also use the CIA (varbind) position number.



Message

Interface Down

* Severity	Critical
* Priority	None
* Lifecycle State	Registered
Source Node	CH-KVI
Source Object	Se0/0/2
Assigned To	

Name	Type	Value
com.hp.ov.nms.apa.symptom	String	IfOperStatusDown
com.hp.ov.nms.apa.symptom_1	String	InterfaceAdminUpC

Updated: 3/29/12 04:56:24 PM Total: 2 Selected: 0 Filter: OFF

Analysis - Incident Summary : InterfaceDown - Message: Interface Down Severity: Critical Lifecycle State: Re

To specify a CIA to use in the identification criteria for duplicate incidents:

1. Navigate to the **Deduplication Comparison Params** form.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Expand the **Incidents** folder.
   c. Select **Syslog MessageConfigurations**.
   d. Do one of the following:
      - To create a new configuration, click the ✱ New icon.
      - To edit an existing configuration, select a row, click the Open icon, and continue.
   e. On the form that opens, navigate to the **Deduplication** tab.
   f. Locate the**Deduplication Comparison Parameters** table.
   g. Do one of the following to specify which CIA:
      - To add a Custom Incident Attribute parameter specification, click the ✱ New icon.
      - To edit an existing Custom Incident Attribute parameter specification, select a row, click the Open icon, and continue.
2. In the Parameter Value field, type (or copy and paste) the exact text string from the Incident form, Custom Attribute tab, **Name** attribute value:

  - NNMi-provided CIA value (see "Custom Incident Attributes Provided by NNMi (Information for Administrators)" on page 647).
  - SNMP trap varbind identified by the Abstract Syntax Notation value (ASN.1).

3. Click **Save and Close** to save your changes and return to the previous configuration form.

Configure Rate (Time Period and Count) for a Syslog Message Incident (HP ArcSight)

For information about each Syslog Message Configuration tab:

Use Rate configuration to track incident patterns *based on the number of incident reoccurrences within a specified time period*. After the count within the specified time period is reached, NNMi emits a Rate Correlation incident and continues to update the Correlation Notes with the number of occurrences within that rate.

Note: Suppression, Enrichment, and Dampening are not supported for Rate incidents.

As long as your defined criteria (Count and Hours, Minutes, Seconds) is sustained, the following information is updated in the Correlation Notes of the Rate Correlation incident:

- the actual number of occurrences of incidents for that sustained rate (Count)
- the sustained time interval (Hours, Minutes, Seconds)

For example, you can set a Rate configuration to track when a link is intermittently down at least three times in 30 minutes. NNMi shows the first occurrence of the rate incident in the incident view and uses Correlation Notes to update the number of incidents and time interval to reflect all the incremental incident occurrences and time periods. To continue the example, if the rate of three times in 30 minutes is sustained for 90 minutes, NNMi updates the Correlation Notes to specify that 9 incidents occurred in 90 minutes.

NNMi provides preconfigured Rate correlations. You can add new Rate correlations.

When you open the Incident form of the newest instance:

- On the General tab, two fields notify you that the Rate correlation is working:
  - **Correlation Nature**: Rate
  - **Count**: x
- On the **Correlated Children** tab, each incident is listed in the table.
- If a Rate Correlation Incident is dampened, note the following:
  - Rate Correlation Incidents inherit the Dampening configuration settings from its Correlated Children.
  - NNMi always retains the Parent Rate Correlation Incident, even if its Child Incidents are Closed and subsequently deleted.

    See "Dampening Incident Configurations" on page 679 for more information about Dampening an incident configuration.

To establish a rate correlation within an incident configuration:

1. Navigate to the **Rate** tab.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Expand the **Incidents** folder.
   c. Select **Syslog Message Configurations**.
   d. Do one of the following:
      - To create a new configuration, click the New icon.
      - To edit an existing configuration, select a row, click the Open icon, and continue.
   e. On the form that opens, locate the **Rate** tab.
2. Provide the definition for this Rate Configuration (see the "Rate Configuration Definition" table).
3. *Optional*. If your Comparison Criteria includes custom incident attributes (CIA) to identify one specific incident, use the Comparison Parameter List table to define each CIA.See "Rate Comparison Parameters Form" on page 678.
4. Click **Save and Close** to save your changes and return to the previous form.

Rate Configuration Definition

Attribute	Description
Enabled	Use this attribute to temporarily disable an incident's rate settings:If enabled, NNMi actively tracks any reoccurrences of the designated incident within the time period you specify, and generates a Rate incident. **Enable** □ = Temporarily disable the selected configuration. **Enable** ☑ = Enable the selected configuration.
Count	Specify the number of reoccurrences required before your Rate Configuration starts working.
Hours	Used with the Minutes and Seconds attributes to specify the time duration within which the reoccurrences are measured.
Minutes	Used with the Hours and Seconds attributes to specify the time duration within which the reoccurrences are measured.
Seconds	Used with the Hours and Minutes attributes to specify the time duration within which the reoccurrences are measured.
Parent Incident	Click the ▾ icon and select Quick Find. Select **Rate Correlation** from the list.
Comparison Criteria	Specify which group of attributes must match before the incident is identified as a duplicate. The possible groups of attributes consist of the following choices. **Name** value of the Incident (from the General tab on the Incident form). **Source Node** value (from the Basics group on the Incident form). Address or

Rate Configuration Definition , continued

Attribute	Description
	name of the node for which the incident was generated. **Source Object** value (from the Basics group on the Incident form). For example, the Source Object for a LinkDown incident is **interface**. **CIA** custom incident attribute values (select from the list displayed on the Custom Attributes tab on the Incident form). If you want to use CIA as part of your comparison criteria, see "Rate Comparison Parameters Form (Syslog Message) (HP ArcSight)" below.
Rate Comparison Parameters	*Optional*. If you selected a Comparison Criteria that includes CIA, you must populate one or more rows in this table. See "Rate Comparison Parameters Form (Management Events)" on page 1204.

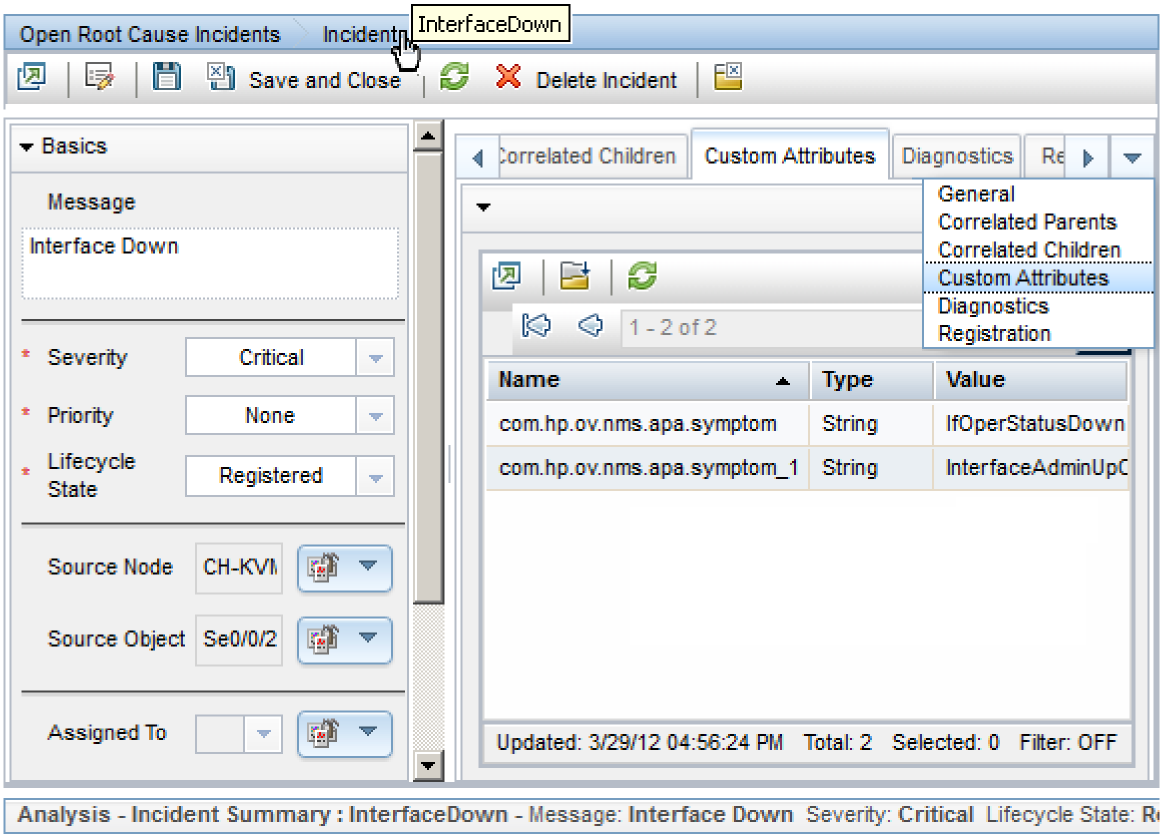
Rate Comparison Parameters Form (Syslog Message) (HP ArcSight)

Custom Incident Attributes (CIAs) are used as parameter values. Parameter values enable accurate identification of duplicate incidents. There are two categories of CIAs:

- SNMP trap varbind values (Name = the MIB varbind identifier, Type = asn_*)
- Custom attributes provided by NNMi (Name = cia.*, Type=String). See "Custom Incident Attributes Provided by NNMi (Information for Administrators)" on page 647.

The group of available CIAs depends on which incident you are configuring for this Rate (for example, CiscoLinkDown). To see which CIAs are available, navigate to an Incident view, select an instance of that incident-type, click the 📂 Open icon and navigate to the Custom Attributes tab. The items listed in the table are the CIAs for that particular incident-type. For example, all CiscoLinkDown incidents would have the same group of CIAs shown in the illustration below.

Note: You can also use the CIA (varbind) position number.



Message: Interface Down

Severity: Critical

Priority: None

Lifecycle State: Registered

Source Node: CH-KVI

Source Object: Se0/0/2

Assigned To:

Name	Type	Value
com.hp.ov.nms.apa.symptom	String	IfOperStatusDown
com.hp.ov.nms.apa.symptom_1	String	InterfaceAdminUpC

Updated: 3/29/12 04:56:24 PM Total: 2 Selected: 0 Filter: OFF

Analysis - Incident Summary : InterfaceDown - Message: Interface Down Severity: Critical Lifecycle State: R

To specify a CIA to use in the identification criteria for duplicate incidents:

1. Navigate to the **Rate Comparison Parameters** form.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Expand the **Incidents** folder.
   c. Select **Syslog Message Configurations**.
   d. Do one of the following:
      - To create a new configuration, click the New icon.
      - To edit an existing configuration, select a row, click the Open icon, and continue.
   e. On the form that opens, navigate to the **Rate** tab.
   f. Locate the **Rate Comparison Parameters** table.
   g. Do one of the following to specify which CIA:
      - To add a Custom Incident Attribute parameter specification, click the New icon.
      - To edit an existing Custom Incident Attribute parameter specification, select a row, click the Open icon, and continue.
2. In the Parameter Value field, type (or copy and paste) the exact text string from the Incident form, Custom Attribute tab, **Name** attribute value:

- NNMi-provided CIA value (see "Custom Incident Attributes Provided by NNMi (Information for Administrators)" on page 647).
- SNMP trap varbind identified by the Abstract Syntax Notation value (ASN.1).

3. Click **Save and Close** to save your changes and return to the previous configuration form.

Configure Actions for a Syslog Message Incident (HP ArcSight)

For information about each Syslog Message tab:

For information about each Actions tab:

You can configure actions to automatically run at any point in the incident lifecycle. For example, you might want to configure an action to occur when an incident of the type you are configuring is generated (**Registered**). When an incident is generated, you might want to automatically open a trouble ticket or send email or page your network operator. After the incident is **Closed**, you might want to automatically close the trouble ticket.

> **Note:** Your actions will not be executed until you enable the Actions configuration by either clicking Enable ☑ on the Actions tab or using the **Actions** → **Enable Configuration** option.

> **Note:** If the NNMi management server is running on a Windows operating system, NNMi runs each action that you configure using the Local System account. If the NNMi management server is running on an HP-UX, Solaris or Linux operating system, NNMi runs each action that you configure using the bin user name. To change the user account associated with actions, see the "Setting the Action Server Name Parameter" section in the *HP Network Node Manager i Software Deployment Reference*.

You can configure actions for incidents generated from SNMP traps, Syslog Messages (HP ArcSight only), Remote NNM 6.x or 7.x events, and the NNMi management events. Any time an incident configuration changes, the action directory is rescanned and any Jython files are reloaded to the NNMi database. See "Lifecycle Transition Action Form (Syslog Message) (HP ArcSight)" on the next page for more information about the actions directory.

Tip: Copy any required Jython files to the NNMi actions directory before you configure an incident action. New or updated actions are loaded into NNMi only when an incident configuration is updated or created. See "Lifecycle Transition Action Form (Syslog Message) (HP ArcSight)" on the next page for the location of the NNMi action directory.

When the action runs, output is logged to the `incidentActions.*.*.log` file. To view the contents of the Actions log, use the **Tools** → **Incident Actions Log** menu option.

See "Verify that NNMi Services are Running" on page 85 for more information about log files and where they are located.

NNMi sets the default values described in the following table.

See the "Maintaining NNMi" chapter in the HP Network Node Manager i Software Deployment Reference for information about changing the default values for Action Server Properties.

Action Server Properties

Property	Description	Value
numProcess	Number of actions that can be run at one time.	10
numJythonThreads	Number of threads the action server uses to run Jython scripts	10
userName	User name under which the action server runs.	bin

To configure an automatic action for an incident:

1. Navigate to the **Actions** tab.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Expand the **Incidents** folder.
   c. Select **Syslog Message Configurations**.
   d. Do one of the following:
      - To create an incident configuration, click the ✳ New icon, and continue.
      - To edit an incident configuration, select a row, click the 📂 Open icon, and continue.
      - To delete an incident configuration, select a row, and click the ✖ Delete icon.
   e. Select the **Actions** tab.
2. From the **Lifecycle Actions** table toolbar, do one of the following:
   - To create an Action configuration, click the ✳ New icon, and continue.
   - To edit an Action configuration, select a row, click the 📂 Open icon, and continue.
   - To delete an Action configuration, select a row, and click the ✖ Delete icon.
3. In the "Lifecycle Transition Action Form (Syslog Message) (HP ArcSight)" below, provide the required information.
4. Click 💾 **Save and Close** to save your changes and return to the previous form.

   The next time the lifecycle changes, NNMi launches the action associated with the lifecycle for the incident of that type.

Lifecycle Transition Action Form (Syslog Message) (HP ArcSight)

For information about each Actions tab:

Use this form to enter the command you want to run when an incident of the type you are configuring is at a particular Lifecycle State. For example, when an incident is generated (**Registered**), you might want to automatically open a trouble ticket or email or page your network operator.

Note: Your actions will not be executed until you enable the Actions configuration by either clicking Enable ☑ on the Actions tab or using the **Actions** → **Enable Configuration** option.

To configure an action for an incidents:

1. Navigate to the **Lifecycle Transition Actions** form:
   a. From the workspace navigation pane, select the **Configuration** workspace.
   b. Expand the **Incidents** folder.
   c. Select **Syslog Message Configurations**.
   d. Select the **Actions** tab.
   e. From the **Lifecycle Transition Action** table toolbar, do one of the following:
      - To create an Action configuration, click the ✱ New icon, and continue.
      - To edit an Action configuration, select a row, click the 📂 Open icon, and continue.
      - To delete an Action configuration, select a row, and click the ✖ Delete icon.
2. Make your configuration choices (see table).

   Note: NNMi reloads the configuration information anytime the incident configuration is changed.
3. Click 💾 **Save and Close** to save your changes and return to the previous form.

Create Action Attributes

Attribute	Description
Lifecycle State	Select a Lifecycle State from the drop-down menu.
Command Type	If you provided a Jython command, select **Jython** from the drop-down list. If you are using an executable or bat file, select **ScriptOrExecutable** from the drop-down list.
Command	Enter one of the following: • A Jython method with the required parameters. • Executable command for the current operating system with the required parameters. When entering a *Command* value, note the following: • Left or right bracket ([ ]) and backtick (` Unicode character: 0060 hex = 96 dec) characters are not permitted in the **Command** attribute. If you need these characters in your shell script, place them in a shell script file and reference that file from the **Command** attribute. • **Windows only**: Shell commands are not permitted in the **Command** attribute. To use shell commands, place them in a shell script file and reference that file from the **Command** attribute. • Verify that you do not have two Jython methods with the same name. Otherwise, NNMi is not able to tell which is the correct method to load. • You can use the same Jython method for more than one incident configuration. • Jython (.py) files must reside in the following directory:

Create Action Attributes, continued

Attribute	Description

	Note: Use absolute paths to executables instead of relying on the PATH variable as it might not be set correctly.
	Windows:
	`%NnmDataDir%\shared\nnm\actions`
	UNIX:
	`/var/opt/OV/shared/nnm/actions`
	• NNMi provides a set of parameters that can pass attribute values from an incident configuration. See "Valid Parameters for Configuring Incident Actions (Management Events)" on page 1216 for more information.

Configure a Payload Filter for an Action (Syslog Message) (HP ArcSight)

The Payload Filter Editor enables you to create expressions that further refine the filters used to select the incidents that cause the configured action to run. Make sure to design any complex Payload Filters offline as a Boolean expression first. This method can help to minimize errors when entering your expressions using the Payload Filter editor.

To create a Payload Filter expression:

1. Navigate to the **Syslog Message Configuration** form:
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Expand the **Incidents** folder.
   c. Select **Syslog Message Configurations**.
   d. Do one of the following:
      i. To create an incident configuration, click the ✳ New icon, and continue.
      ii. To edit an incident configuration, select a row, click the 📂 Open icon, and continue.
      iii. To delete an incident configuration, select a row, and click the ✖ Delete icon.
2. Select the **Actions** tab.
3. Do one of the following:
   a. To create a new configuration, click the ✳ New icon.
   b. To edit an existing configuration, select a row, click the 📂 Open icon, and continue.
4. Select the **Payload Filter** tab.
5. Define your Payload Filter (see table). Also see "Guidelines for Creating a Payload Filter".

a. Plan out the logic needed for your Filter String.

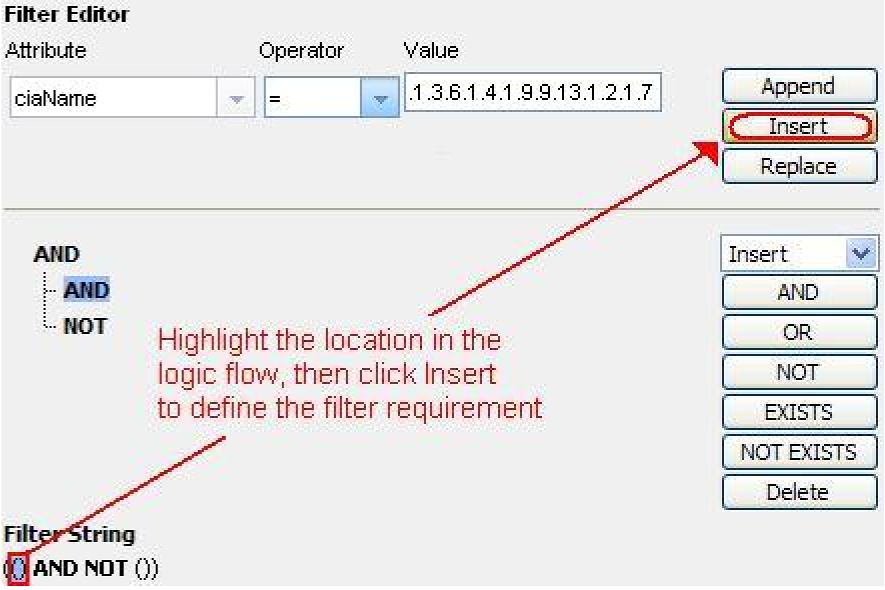
b. Use the buttons on the bottom half of the Additional Filters Editor to establish the logic structure.

   For example, to establish the following structure, select **Insert**, then click **AND**, then **NOT**, and then **AND** a second time:

   ```
   (( ) AND NOT ( ))
   ```

c. Now place your cursor in a location within the displayed Filter String, and use the top half of the filter editor to define the parameters of the highlighted filter requirement.

   For example, select a set of parentheses and use the Insert button to specify the filter requirement within those parentheses:



Attribute: ciaName

Operator: =

Value: .1.3.6.1.4.1.9.9.13.1.2.1.7

Highlight the location in the logic flow, then click Insert to define the filter requirement

Filter String
(() AND NOT ())

6. Click **Save and Close**.

7. Click **Save and Close** to save your changes and return to the previous form.

When creating a Payload Filter, note the following:

- Payload Filter expressions for the `like` and `not like` operators use the syntax defined for java regular expressions (java.util.regex Pattern class)
- You must use a `ciaName` that already exists in the trap or event you are configuring.
- Each set of expressions associated with a Boolean Operator is treated as if it were enclosed in parentheses and evaluated together.
- View the expression displayed under **Filter String** to see the logic of the expression as it is created.
- The `AND` and `OR` Boolean Operators must contain at least two expressions as shown in the example below.

The following example filters incidents on voltage state. Using this Payload Filter, you could then configure the Basics settings of the Enrichment Configuration to set the severity and message format to all incidents that return a state value of `4` or `5`.

```
OR
  ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7
   ciaValue = 4
  AND
   ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7
   ciaValue = 5
```

NNMi evaluates the expression above as follows:

```
(ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7 AND ciaValue = 4) OR (ciaName
= .1.3.6.1.4.1.9.9.13.1.2.1.7 AND ciaValue = 5)
```

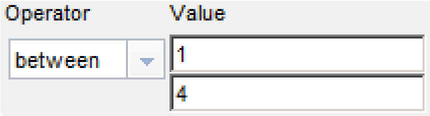
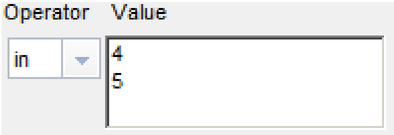
NNMi finds all incidents with a varbind value of `.1.3.6.1.4.1.9.9.13.1.2.1.7` and CIA value of **4** or **5**.

- The placement of your cursor and the subsequent text that is selected is important when performing operations using the Payload Filter Editor. For example, you append to, replace, or change the indentation of the expression that is selected.
- The placement of your cursor and the subsequent text that is selected is especially important when adding your Boolean operators.

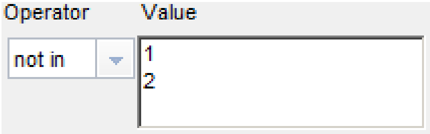
Payload Filter Editor Components

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: • ciaName • ciaValue
Operator	Valid operators are described below. • **=** Finds all values equal to the value specified. Click here for an example. Example: `ciaName=.1.3.6.1.4.1.9.9.13.1.2.1.7` matches any incident that contains a varbind with the name value **.1.3.6.1.4.1.9.9.13.1.2.1.7**. • **!=** Finds all values not equal to the value specified. Click here for an example. Example: `ciaName!=.1.3.6.1.4.1.9.9.13.1.2.1.7` matches any incident that contains a varbind with the name value other than **1.3.6.1.4.1.9.9.13.1.2.1.7**. • **<** Finds all values less than the value specified. Click here for an example. Example: `ciaValue < 6` matches any incident that contains a varbind value less than **6**.

Payload Filter Editor Components, continued

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: • ciaName • ciaValue
	• **<=** Finds all values less than or equal to the value specified. Click here for an example. Example: `ciaValue <= 6` matches any incident that contains a varbind value less than or equal to **6**. • **>** Finds all values greater than the value specified. Click here for an example. Example: `ciaValue > 4` matches any incident that contains a varbind value greater than **4**. • **>=** Finds all values greater than or equal to the value specified. Click here for an example. Example: `ciaValue >= 4` matches any incident that contains a varbind with values greater than or equal to **4**. • **between** Finds all traps or events that include a varbind value equal to and between the two values specified. Click here for an example. Example: `ciaValue between`  Operator: between Value: 1 4 matches any incident that contains a varbind value equal to or greater than **1** and equal to or less than **4**. **Note**: As shown in the example, each value must be entered on a separate line. • **in** Finds any match to at least one value in a list of values. Click here for an example. Example: `ciaValue in`  Operator: in Value: 4 5 matches any incident that contains a varbind value of either **4** or **5**. **Note**: As shown in the example, each value must be entered on a separate line. NNMi displays the list of attributes using comma-separated values enclosed in

Payload Filter Editor Components, continued

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: • ciaName • ciaValue
	parentheses, for example (**4, 5**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line. • **is not null** Finds all non-blank values. Click here for an example. Example: `ciaValue is not null` matches any incident with varbind values. • **is null** Finds all blank values. Click here for an example. Example: `ciaValue is null` matches any incident with no varbind values. • **like** Finds matches using wildcard characters. Click here for more information about using wildcard characters. The period asterisk (.*) characters mean *any number of characters of any type at this location*. The period (.) character means *any single character of any type at this location*. **Note**: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below. Examples: `ciaName like  \Q .1.3.6.1.4.1.9.9\E.*` finds all traps or events that contain varbind names that begin with **.1.3.6.1.4.1.9.9** and (optionally) end with any number of characters. `ciaValue like .*Chicago.*` finds all traps or events that contain a varbind value that includes the string **Chicago**. • **not between** Finds all values except those between the two values specified. Click here for an example. Example: `ciaValue not between 5 8` matches an incident that contains a varbind with the values less than **5** or greater than **8**. • **not in** Finds all values except those included in the list of values. Click here for an example. Example: `ciaValue not in`  Operator: not in Value: 1 2

Payload Filter Editor Components, continued

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: • ciaName • ciaValue
	matches any incident that contains a varbind with values other than **1** and **2**. **Note**: As shown in the example, each value must be entered on a separate line. NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example, (**1, 2**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line. • **not like** Finds all that do not have the values specified (using wildcard strings). Click here for an example. The period asterisk (.*) characters mean *any number of characters of any type at this location*. The period (.) character means *any single character of any type at this location*. **Note**: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below. Example: `ciaName not like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that does not begin with **`.1.3.6.1.4.1.9.9`**. `ciaValue not like .*Chicago.*` finds all traps or events that do not contain a varbind value that includes the string **Chicago**.
Value	The value for which you want NNMi to search. Note the following: • The values you enter are case sensitive. • NNMi displays a variable number of value fields depending on the Operator selected. For example, the `between` Operator causes two value fields to be displayed. • The `between, in` and `not in` operators require that each value be entered on a separate line.

Payload Filter Editor Buttons

Button	Description
Append	Appends the current expression (Attribute, Operator,and Value) to the selected expression already included in the filter string.
Insert	Inserts the current expression (Attribute, Operator,and Value) in front of the

Payload Filter Editor Buttons, continued

Button	Description
	cursor location within the Filter String.
Replace	Replaces the selected expression with the expression displayed in the Attribute, Operator, and Value fields.
AND	Inserts the AND Boolean Operator in the selected cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
OR	Inserts the OR Boolean Operator in the current cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
NOT	Can be used in any part of the Filter String to specify that NNMi should exclude interfaces with values that pass the expression that immediately follows the `NOT`. For example, when evaluating the following expression, NNMi includes interfaces with (interface description) `ifDesc` containing **VLAN**, and excludes any Interfaces that have **VLAN10** for the (interface name) `ifName` value: `(ifDesc like VLAN AND NOT (ifName=VLAN10))` **Note**: View the expression displayed under Filter String to see the logic of the expression as it is created .
EXISTS	Used for filters that include Capabilities or Custom Attribute names and values in the Filer String. Indicates that you want NNMi to consider interfaces that do not have any Capabilities or Custom Attributes when evaluating the Filter String. **Note**: If you include Capabilities or Custom Attribute names and values in the Filer String, but do not use **EXISTS** or **NOT EXISTS**, NNMi excludes from its search interfaces that do not include Capabilities or Custom Attributes. For example, when evaluating the following Filter String, NNMi includes interfaces with (interface description) `ifDesc` containing **VLAN**, as well as any Interfaces Custom Attribute Role value is **LAN Connection to Oracle Server**: `(ifDesc like VLAN OR EXISTS((customAttrName=Role AND customAttrValue=LAN Connection to Oracle Server)))` **Note**: View the expression displayed under Filter String to see the logic of the expression as it is created.
NOT EXISTS	Used for filters that include Capabilities or Custom Attribute names and values in the Filer String. Indicates that you want NNMi to consider interfaces that do not have any Capabilities or Custom Attributes when evaluating the Filter String, but exclude the interfaces that match the expression that follows the **NOT EXISTS**.

Payload Filter Editor Buttons, continued

Button	Description
	Note: If you include Capabilities or Custom Attribute names and values in the Filer String, but do not use **EXISTS** or **NOT EXISTS**, NNMi excludes from its search interfaces that do not include Capabilities or Custom Attributes. For example, when evaluating the following expression, NNMi includes interfaces with (interface description) `ifDesc` containing **VLAN**, and excludes any Interfaces that have the Custom Attribute Role and that Role value is **LAN Connection to Oracle Server**: `(ifDesc like VLAN OR NOT EXISTS((customAttrName=Role AND customAttrValue=LAN Connection to Oracle Server)))` **Note**: View the expression displayed under Filter String to see the logic of the expression as it is created.
Delete	Deletes the selected expression. **Note**: If the Boolean Operator is selected, the Payload Filter Editor deletes all expressions associated with the Boolean Operator.

Valid Parameters for Configuring Incident Actions (Syslog Message) (HP ArcSight)

When configuring incident actions, consider using incident information as part of the action. NNMi provides the following parameter values. Use these parameters as variables in your Jython or executable files.

Tip: See the Using the Incident Form for more information about the parameter values.

Note: NNMi stores varbind values as custom incident attributes (CIAs).

Tip: If a value is not stored for a parameter, it is returned as “null”.

See "Lifecycle Transition Action Form" on page 748 for more information about configuring incident actions.

Valid Parameters Visible From an Incident's Form

Parameter Value	Description
$category, $cat	Value of the Category attribute in the Incident form.
$count, $cnt	Value representing the number of Custom Incident Attributes that appear in the Incident form.
$family, $fam	Value from the Family attribute in the Incident form.
$firstOccurrenceTime, $fot	Value from the First Occurrence Time attribute in the incident form.
$lastOccurrenceTime, $lot	Value from the Last Occurrence Time attribute in the incident form.

Valid Parameters Visible From an Incident's Form, continued

Parameter Value	Description
$lifecycleState, $lcs	Value from the Lifecycle State attribute in the Incident form.
$name	Value of the Name attribute from the incident configuration.
$nature, $nat	Value from the Nature attribute in the Incident form.
$origin, $ori	Value from the Origin attribute in the Incident form.
$originOccurrenceTime, $oot	Value from the Origin Occurrence Time attribute in the incident form.
$priority, $pri	Value from the Priority attribute in the Incident form.
$severity, $sev	Value of the Severity attribute of the Incident form.

Valid Parameters Visible from a Node Form

Parameter Value	Description
$managementAddress, $mga	Value from the Management Address attribute of the incident's source Node's form or SNMP Agent form.
$otherSideOfConnectionManagementAddress, $oma	If the incident's Source Node is part of a Layer 2 Connection, this attribute is the value of the Management Address of a node on the other side of the Layer 2 Connection.
$sourceNodeLongName, $sln	The fully-qualified DNS name as displayed in the Hostname attribute of the incident's source Node's form.
$sourceNodeName, $snn	Value from the Name attribute of the incident's source Node's form.
$sysContact, $sct	Value from the System Contact attribute of the incident's source Node form: General tab.
$sysLocation, $slc	Value from the System Location attribute of the incident's source Node form: General tab.

Valid Parameters Visible from an Interface Form

Parameter Value	Description
$ifAlias, $ifa	Value from the IfAlias attribute for the interface that is the incident's source object.
$ifConfigDupSetting,	Configured Duplex Setting on the port associated with the interface that

Valid Parameters Visible from an Interface Form , continued

$icd	is the incident's source object.
$ifDesc, $idc	Value from the ifDesc attribute for the interface that is the incident's source object.
$ifIndex, $idx	Value from the ifIndex attribute for the interface that is the incident's source object.
$ifIpAddr, $iia	IP Address values associated with the interface that is the incident's source object. If multiple IPaddresses are associated with the interface, this parameter returns a comma-separated list.
$ifName, $ifn	Value from the ifName attribute for the interface that is the incident's source object.
$ifPhysAddr, $ipa	Value from the Physical Address attribute for the interface that is the incident's source object.
$ifSpeed, $isp	Value from the ifSpeed attribute for the interface that is the incident's souce object.
$ifType, $itp	Value from the ifType attribute for the interface that is the incident's souce object.

Valid Parameters Visible from a Layer 2 Connection Form

Parameter Value	Description
$otherSideOfConnectionConfigDupSetting, $ocd	If the incident's source Node is part of a Layer 2 Connection, this parameter contains the Configured Duplex Setting on the port associated with the interface on the other side of the connection.
$otherSideOfConnectionIfAlias, $oia	If the incident's Source Node is part of a Layer 2 Connection, this parameter is the value of the ifAlias of one of the interfaces on the other side of the Layer 2 Connection.
$otherSideOfConnectionIfDesc, $odc	If the incident's Source Node is part of a Layer 2 Connection, this parameter contains the ifDescr attribute value for the interface on the other side of the Layer 2 Connection.
$otherSideOfConnectionIfIndex, $odx	If the incident's Source Node is part of a Layer 2 Connection, this parameter contains the ifIndex attribute value for the interface on the other side of the connection.
$otherSideOfConnectionIfName, $ofn	If the incident's Source Node is part of a Layer 2 Connection, this parameter contains the ifName attribute value for the interface on the other side of the connection.

Valid Parameters Visible from a VLAN Form

Parameter Value	Description
$impVlanIds, $ivi	Value from the VLAN Id attribute associated with the interface that is the incident's source object. To access this information from an interface form, navigate to the VLAN Port tab and open the form for the VLAN of interest. If the interface is part of more than one VLAN, this parameter returns a comma-separated list.
$impVlanNames, $ivn	Value from the Global VLAN Name attribute associated with the interface that is the incident's source object. To access this information from a Node form or Interface form, navigate to the VLAN Ports tab. If the node or interface is part of more than one VLAN, this parameter returns a comma-separated list.

Valid Parameters Not Visible From a Form

Parameter Value	Description
$id	Unique Object Identifier attribute value for the incident (unique across the entire NNMi Database).
$firstOccurrenceTimeMs, $fms	Value from the First Occurrence Time attribute in the incident form, converted to millseconds (measured since January 1, 1970, 00:00:00 GMT - Greenwich Mean Time).
$lastOccurrenceTimeMs, $lms	Value from the Last Occurrence Time attribute in the incident form, converted to millseconds (measured since January 1, 1970, 00:00:00 GMT - Greenwich Mean Time).
$messageFormat, $msg	*Valid for Incident actions only*. Message text displayed for an incident when this parameter is included as an argument to an incident action.
$oid	Value of the unique object identifier (oid) for the incident configuration that originated from either an SNMP trap, Management Event, or Remote NNM 6.x or 7.x event.
$otherSideOfConnection, $osc	If the incident's Source Node is part of a Layer 2 Connection, this attribute is the following combination of values for the node and one of its interfaces on the other side of the Layer 2 Connection: The fully-qualified DNS name of the node appended with the interface Name in the following format: <*fully-qualified DNS name*> [*interface_name*]
$originOccurrenceTimeMs, $oms	Value from the Origin Occurrence Time attribute in the incident form, converted to millseconds (measured since January 1, 1970, 00:00:00 GMT - Greenwich Mean Time).

Valid Parameters Not Visible From a Form, continued

Parameter Value	Description
$sourceNodeUuid, $snu	Universally Unique Object Identifier attribute value of the source node object for the incident (unique across all databases). This identifier distinguishes the source node object instance from all other node objects.
$sourceObjectClass, $soc	Value of the object class for the object you want to include. Use this parameter to request more details of a class of objects through a web service. Examples of object classes include: `com.hp.ov.nms.model.core.Interface` and `com.hp.ov.nms.model.snmp.SnmpAgent`.
$sourceObjectName, $son	Value from the Name attribute of the source object. For example, an interface object is named according to the MIB ifName. Each ifName varies according to the vendor's conventions. Using the name `4/1` as an example, `4` represents `the board number and 1` represents the port number.
$sourceObjectUuid, $sou	Universally Unique Object Identifier attribute value of the source object for the incident (unique across all databases). This identifier distinguishes the source object instance from all other similar object instances..
$uuid	Universally Unique Object Identifier attribute value of the incident (unique across all databases). This identifier distinguishes the incident object instance from all other incident objects.

Valid Parameters Established in Custom Incident Attributes

Parameter Value	Description
$<position_ number>	Value of the custom incident attribute (CIA) position number for any CIA that originated from a varbind or was added by NNMi. For example, to indicate you want to use the varbind in position 1, enter: `$1` NNMi stores varbind values as Custom Incident Attributes. If you know the varbind position number, use this parameter.
$<CIA_ name>	Value of the name that is used for the custom incident attribute. For example, `$mycompany.mycia`. NNMi provides CIA values for configuring Management Events. See Custom Incident Attributes Provided by NNMi for more information about custom incident attributes.
$<CIA_ oid>	Value of the object identifier for any custom incident attribute that originated as a varbind. For example, `$.1.3.6.1.6.3.1.1.5.1`. Use this parameter when you are not certain of a custom incident attribute (varbind) position number.
$*	Used to indicate you want all of the custom incident attribute values originating as varbinds, to be passed to the action configuration. Each varbind is returned in the

Valid Parameters Established in Custom Incident Attributes, continued

Parameter Value	Description
	following format: `$<CIA_name>:<CIA_value>` in which the custom incident attribute name appears followed by the custom incident attribute value.

The function described in the following table replaces the specified numeric value with the associated text value stored in the CIA.

Note: The associated MIB must have been loaded using the nnmloadmib.ovpl command.

Functions to Generate Values Within Incident Messages

Function	Description
$text ($<position_number>)	The <*position_number*> argument specifies the numeric value of the custom incident attribute (CIA) position number for any CIA that originated from a varbind or was added by NNMi. For example, to indicate you want to use the varbind in position 1, enter: `$1`. After the function runs, NNMi replaces the numeric value with the text value stored in the CIA. **Note**: If a text value is not available, NNMi returns the numeric value.
$text ($<CIA_oid>)	The <CIA_oid> argument specifies the object identifier for any custom incident attribute that originated as a varbind. For example, `$.1.3.6.1.6.3.1.1.5.1`. Use this argument to the $text function when you are not certain of a custom incident attribute (varbind) position number. After the function runs, NNMi replaces the numeric value with the text value stored in the CIA. **Note**: If a text value is not available, NNMi returns the numeric value.

Configure Management Events

Management events are those events that are generated from the NNMi Causal Engine. You can configure how you want these events to be displayed in the incident views provided by NNMi. The types of things you configure include its name, category, and the format of its message.

> **Note**: Custom created Management Incidents are for use in Custom Correlations. See "Configure a Correlation Rule" on page 682 and "Configure a Causal Rule" on page 714 for more information.

To configure a management event:

1. Navigate to the **Management Events Configuration** form.
   a. From the workspace navigation panel, select the **Configuration** workspace.

   b. Expand the **Incidents** folder.

   c. Select **Management Event Configurations**.

2. Do one of the following:

   a. To create a management event configuration, click the * New icon, and continue.

   b. To edit a management event configuration, double-click the row representing the configuration you want to edit, and continue.

   c. To delete a management event configuration, select a row, and click the ✖ Delete icon.

3. In the Management Event Configuration form, provide the required information.

4. Click **Save and Close** to save your changes and return to the **Incident Configuration** form.

The next time that a management event of this type arrives into the database, NNMi creates an associated incident to display in the appropriate console incident views.

Management Event Form

To configure incidents originating from management events:

1. Navigate to the **Management Event Configuration** form:

   a. From the workspace navigation pane, select the **Configuration** workspace.

   b. Expand the **Incidents** folder.

   c. Select **Management Event Configurations**.

2. Make your configuration choices (see table).

   Note: If you want to add or edit a management event configuration, verify that **Enabled** ☑ is selected.

   a. To add a management event configuration, click the * New icon, and continue.

   b. To edit a management event configuration, double-click the row representing the configuration you want to edit, and continue.

   c. To delete a management event configuration, click the ✖ Delete icon.

3. Click **Save and Close** to save your changes and return to the previous form.

Tasks for Management Event Configuration

Task	How
"Specify the Incident Configuration Name (Management Events)" on page 1084	Use the **Basics** group of the **Management Event Configuration** form. Specify a name that helps you to identify the configuration for subsequent use.
Specify whether you want to enable this configuration.	In the **Basics** group of the **Management Event Configuration** form, verify that **Enable** ☑ is selected for each configuration you want to use.

Tasks for Management Event Configuration, continued

Task	How
"Specify Category and Family Attribute Values for Organizing Your Incidents (Management Events)" on page 1084	Use the **Basics** group of the **Management Event Configuration** form. You can organize your incidents using Category and Family.
"Specify the Incident Severity (Management Events)" on page 1089	Use the **Basics** group of the **Management Event Configuration** form. Possible Severity values include: **Normal, Warning, Minor, Major,** and **Critical**.
"Specify Your Incident Message Format (Management Events)" on page 1089	Use the **Basics** group of the **Management Event Configuration** form. The message format determines the message to be displayed for the incident.
"Specify a Description for Your Incident Configuration (Management Events)" on page 1097	Use the **Basics** group of the **Management Event Configuration** form. Provide a meaningful description.
Specify an Author for Your Incident Configuration (Management Events)	Use the **Basics** pane of the **Management Event Configuration** form to indicate who created or last modified the event. **Caution:** If the Author attribute value is **HP Network Node Manager**, any changes are at risk of being overwritten in the future. • Click **Lookup** and select **Show Analysis** to display details about the currently selected Author. • Click **Quick Find** to access the list of existing Author values. • Click **New** to create an Author value.

After you complete the Basic Configuration for the management event, you can also choose to configure the information described in the following table.

Additional Configurations

Task	How
"Correlate Duplicate Incidents (Deduplication Configuration)" on page 659	Select the **Deduplication** tab to specify duplicate incidents that you want to be suppressed.
"Track Incident Frequency (Rate: Time Period and Count)" on page 659	Select the **Rate** tab to specify a rate for duplicate incidents. After the rate limit is reached, NNMi generates an Incident to notify you of the problem.

Additional Configurations, continued

Task	How
"Configure an Action for an Incident" on page 748	Select the **Actions** tab to specify actions that should occur automatically when an incident changes its Lifecycle State.
"Configure Diagnostics for an Incident (NNM iSPI NET)" on page 757	Select the **Node Settings** tab to specify diagnostic actions that should occur automatically when an incident reaches a selected Lifecycle State for a node that belongs to a particular Node Group.

Configure Basic Settings for a Management Event Incident

The Basics settings for a Remote NNM 6.x/7.x event incident specifies general information for an incident configuration, including the name, severity, and message.

Note: In the **Basics** group of the **Management Event Configuration** form, verify that **Enable** ☑ is selected for each configuration you want to use.

For information about each Management Events tab:

To configure Basic settings for a Management Event incident:

Navigate to the **Management Event Configuration** form:

1. From the workspace navigation panel, select the **Configuration** workspace.
2. Expand the **Incidents** folder.
3. Select **Management Event Configurations**.
4. Do one of the following:
    a. To create an incident configuration, click the New icon, and continue.
    b. To edit an incident configuration, select a row, click the Open icon, and continue.
    c. To delete an incident configuration, select a row, and click the Delete icon.
5. Configure the required Basic settings (see the Basic Attributes table).
6. Click **Save and Close** to save your changes and return to the previous form. NNMi uses the SNMP Object ID to enable forwarding of Management Events as SNMP traps. NNMi automatically assigns a unique SNMP Object ID to all Management Events provided by NNMi.

Basic Attributes for SNMP Trap Configuration

Task	How
"Specify the Incident Configuration Name (Management Events) " on page 1084	Use the **Basics** pane of the **Management Event Configuration** form. Specify a name that helps you to identify the configuration for subsequent use.
SNMP Object ID	The SNMP Object ID assigned by NNMi.

Basic Attributes for SNMP Trap Configuration, continued

Task	How
	Note the following about the SNMP Object ID that appears in the Basics settings of the Management Event Configuration form: • The Management Event SNMP Object ID is used when sending Management Events to another application. For example, you might want to send NNMi Management Event to an event consolidator such as HP Operations Manager. The SNMP Object ID is used to uniquely identify the management event in the application receiving the event. • NNMi assigns a unique SNMP Object ID to each Management Event configuration it provides. If you choose to create a new Management Event configuration, NNMi assigns the following "generic" SNMP Object ID to these user-created configurations: `.1.3.6.1.4.1.11.2.17.19.2.0.9999` • For user-defined Management Events, a combination of the SNMP Object ID and the user-defined event name must be used to uniquely identify the Management Event in an application receiving the event. • See the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals` for more information. • If you choose to create a new Management Event configuration, NNMi automatically assigns the same "generic" SNMP Object ID to all new Management Event configurations.
Specify whether you want to enable this configuration.	In the **Basics** group of the **Management Event Configuration** form, verify that **Enable** ☑ is selected for each configuration you want to use.
"Specify Category and Family Attribute Values for Organizing Your Incidents (Management Events) " on page 1084	Use the **Basics** pane of the **Management Event Configuration** form. You can organize your incidents using Category and Family.
"Specify the Incident Severity (Management Events) " on page 1089	Use the **Basics** pane of the **Management Event Configuration** form. Possible Severity values include: **Normal, Warning, Minor, Major,** and **Critical**.
"Specify Your Incident Message Format (Management Events) " on page 1089	Use the **Basics** pane of the **Management Event Configuration**form. The message format determines the message to be displayed for the incident.

Basic Attributes for SNMP Trap Configuration, continued

Task	How
"Specify a Description for Your Incident Configuration (Management Events) " on page 1097	Use the **Basics** pane of the **Management Event Configuration** form. Provide a meaningful description.
Specify an Author for Your Incident Configuration (Management Events)	Use the **Basics** pane of the **Management Event Configuration** form to indicate who created or last modified the event. **Caution:** If the Author attribute value is **HP Network Node Manager**, any changes are at risk of being overwritten in the future. • Click **Lookup** and select **Show Analysis** to display details about the currently selected Author. • Click **Quick Find** to access the list of existing Author values. • Click * **New** to create an Author value.

After you complete the Basic Configuration for the remote NNM 6.x/7.x event, you can also choose to configure the information described in the following table.

Additional Incident Configurations

Task	How
"Configure Interface Settings for a Management Event Incident" on page 1098	Select the **Interface Settings** tab to specify an Interface Group to which you want your incident configuration to apply.
"Configure Node Settings for a Management Event Incident" on page 1135	Select the **Node Settings** tab to specify a Node Group to which you want your incident configuration to apply.
"Configure Suppression Settings for a Management Event Incident" on page 1173	Select the **Suppression** tab to specify the criteria for discarding incidents that match the selected incident configuration.
"Configure Enrichment Settings for a Management Event Incident" on page 1181	Select the **Enrichment** tab to specify enhancements for the selected incident configuration.
"Configure Dampening Settings for a Management Event Incident" on page 1185	Select the **Dampen** tab to specify the time interval that must be met before the incident appears in an Incident view.
"Configure Deduplication for a Management Event Incident" on page 1194	Select the **Deduplication** tab to specify duplicate incidents that you want to be suppressed.

Additional Incident Configurations, continued

Task	How
"Configure Rate (Time Period and Count) for a Management Event Incident" on page 1202	Select the **Rate** tab to specify a rate for duplicate incidents. After the rate limit is reached, NNMi generates an Incident to notify you of the problem.
"Configure Actions for a Management Event Incident" on page 1206	Select the **Actions** tab to specify actions that should occur automatically when an incident changes its Lifecycle State.

Specify the Incident Configuration Name (Management Events)

When providing the Name for an incident configuration, use the following guidelines:

Name

The name is used to identify the incident configuration and must be unique. Use a name that will help you to remember the purpose or kind of management event, remote NNM 6.x or 7.x event or SNMP trap, for which you are configuring an incident. Name is also used to identify your Pairwise configurations.

Alpha-numeric and special characters (~ ! @ # $ % ^ & * ( ) _+ -) are permitted. No spaces are permitted.

Specify Category and Family Attribute Values for Organizing Your Incidents (Management Events)

When configuring incidents, NNMi provides the Category and Family attributes to help you organize your incidents.

Preconfigured Categories
The Category attribute helps you organize your incidents. Select the category that you want to be associated with this type of incident when it appears in an incident view. Each of the possible Category values is described in the following table.

Incident Categories Provided by NNMi

Category	Description
Accounting	Used to indicate problems related to usage statistics and allocation of costs associated with the billing of time and services provided by devices. This category is not used by NNMi with default configurations, but it is available for incidents you define.
Application Status	Indicates there is a problem with the health of the NNMi software. Examples of these kinds of events include license expiration (see "Extend a Licensed

Incident Categories Provided by NNMi, continued

Category	Description
	Capacity" on page 1575) or that a certain NNMi process or service lost connection to the Process Status Manager (see "Stop or Start an NNMi Process" on page 82 and "Stop or Start NNMi Services" on page 86).
Configuration	Indicates there is a problem with the configuration of a managed device. For example, there is a physical address mismatch.
Fault	Indicates a problem with the network, for example Node Down.
Performance	Indicates a Monitored Attribute value *crossed* a configured threshold. For example, Disk Space Utilization exceeds the configured threshold criteria for High Value = 90 percent .
Security	Indicates there is a problem related to authentication. For example, an SNMP authentication failure.
Status	Indicates some kind of status message. Examples of these kinds of incidents include "SNMP Link Up" or an "HSRP Group status Normal" message.

Note: You can add your own Category entries to NNMi. See "Create an Incident Category (Management Events)" on page 1087 for more information.

You can use **Family** attribute values to further categorize the types of incidents that might be generated. Each of the possible values are described in the following table.

Incident Family Attribute Values Provided by NNMi

Family	Description
Address	Indicates the incident is related to an address problem.
Aggregated Port	Indicates the incident is related to a **Link Aggregation**[1] problem.
BGP	Indicates the incident is related to a problem with BGP (Border Gateway Protocol). This family is not used by NNMi with default configurations, but it is available for incidents you define.
Board	Indicates the incident is related to a board problem. This family is not used by NNMi with default configurations, but it is available for incidents you define.
Chassis	Indicates the incident is related to an board problem. This family is not used by NNMi with default configurations, but it is available for incidents you define.

[1] Protocols used on Switches to configure multiple Interfaces (Aggregation Member Interfaces) to function as if they were one (an Aggregator Interface). When two Aggregator Interfaces establish a connection, that connection is an Aggregator Layer 2 Connection. The Aggregator Layer 2 Connection appears on Layer 2 Neighbor View maps as a thick line with an Interface icon at each end (representing the Aggregator Interface). When you double-click the thick line, it converts into multiple thin lines representing the participating Aggregation Member Layer 2 Connections with their Aggregation Member Interfaces at each end of the lines.

Incident Family Attribute Values Provided by NNMi, continued

Family	Description
Component Health	Indicates the incident is related to Node Component metrics collected by NNMi. See "Node Form: Node Component Tab" for more information about the Node Component metrics collected.
Connection	Indicates the incident is related to a problem with one or more connections.
Correlation	Indicates the incident has additional incidents correlated beneath it. These incidents are associated with a duplicate count so that you can determine the number of correlated incidents associated with it.
Custom Poller	Indicates the incident is related to the NNMi Custom Poller feature. See "About Custom Poller".
HSRP	*NNMi Advanced*. Indicates the incident is related to a problem with Hot Standby Router Protocol (**HSRP**[1]).
Interface	Indicates the incident is related to a problem with one or more interfaces.
License	Indicates the incident is related to a licensing problem.
NNMi Health	Indicates the incident is related to NNMi Health. See the Check NNMi Health for more information.
Node	Indicates the incident is related to a node problem.
OSPF	Indicates the incident is related to an OSPF problem. This family is not used by NNMi with default configurations, but it is available for incidents you define.
RAMS	*NNMi Advanced*. Indicates the incident is related to a Router Analytics Management System problem.
RMON	Indicates the incident is related to a Remote Monitor (IETF standard, RFC 1757) problem. This family is not used by NNMi with default configurations, but it is available for incidents you define.
RRP	*NNMi Advanced*. Indicates the incident is related to a problem with a Router Redundancy Protocol configuration.
STP	Indicates the incident is related to Spanning-Tree Protocol problem. This family is not used by NNMi with default configurations, but it is available for incidents you define.
Syslog	NNMi does not use this Family with default configurations. It is available for incidents you define.
Trap Analysis	Indicates the incident is related to an SNMP trap storm.

[1] Hot Standby Router Protocol

Incident Family Attribute Values Provided by NNMi, continued

Family	Description
VLAN	Indicates the incident is related to a problem with a virtual local area network.
VRRP	*NNMi Advanced*. Indicates the incident is related to a problem with Virtual Router Redundancy Protocol (**VRRP**[1]).

Note: You can add your own Family entries to NNMi. See "Create an Incident Family (Management Events)" on the next page for more information.

Create an Incident Category (Management Events)

The Category attribute helps you organize your incidents. Create any Category that makes sense to you and your team. For a list of the Category codes provided by NNMi, "Specify Category and Family Attribute Values for Organizing Your Incidents (Management Events)" on page 1084.

To create a new incident Category:

1. Navigate to the **Management Event Configuration Category** form.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Expand the **Incidents** folder.
   c. Select **Management Event Configurations**.
   d. Do one of the following:
      - To create an incident configuration, click the New icon, and continue.
      - To edit an incident configuration, select a row, click the Open icon, and continue.
      - To delete an incident configuration, select a row, and click the Delete icon.
   e. In the configuration form, locate the **Category** attribute.
   f. Click the Lookup icon, and select New.
2. Provide the required information (see table).
3. Click **Save and Close** to save your changes and return to the previous form.

Category Code Attributes

Name	Description
Label	Incident category name. For example, Hardware Faults, or Cisco Error. Maximum size is 255 characters. Alpha-numeric, spaces, and underline characters are permitted.
Unique Key	**Caution**: After you click **Save and Close**, this value cannot be changed. Used as a unique identifier when exporting and importing configuration definitions. To ensure that the value you enter is unique, it is recommended that you use the Java name space convention when providing this value. It is also useful to include the label

[1] Virtual Router Redundancy Protocol

Category Code Attributes , continued

Name	Description

	value as part of the unique key as shown in the following examples:
	`com.<your_company_name>.nnm.trapConf.category.<category_label>`
	`com.<your_company_name>.nnm.eventConf.category.<category_label>`
	`com.<your_company_name>.nnm.inciConf.category.<category_label>`
	The maximum length is 80 characters. Alpha-numeric characters and periods are permitted. Spaces are not permitted.

Create an Incident Family (Management Events)

The Family attribute helps you organize your incidents. Create any Family that makes sense to you and your team. For a list of the Family codes provided by NNMi, "Specify Category and Family Attribute Values for Organizing Your Incidents (Management Events)" on page 1084.

To create a new incident Family:

1. Navigate to the **Incident Family** form.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Expand the **Incidents** folder.
   c. Select **Management Event Configurations** .
   d. Do one of the following:
      - To create an incident configuration, click the New icon, and continue.
      - To edit an incident configuration, select a row, click the Open icon, and continue.
      - To delete an incident configuration, select a row, and click the Delete icon.
   e. In the configuration form, locate the **Family** attribute.
   f. Click the Lookup icon, and select New.
2. Provide the required information (see table).
3. Click **Save and Close** to save your changes and return to the previous form.

Family Attributes

Name	Description
Label	Family name. For example, Hardware Faults, or Cisco Error. Maximum size is 255 characters. Any character type is valid.
Unique Key	**Caution**: After you click **Save and Close**, this value cannot be changed.

Family Attributes , continued

Name	Description

	Used as a unique identifier when exporting and importing configuration definitions. To ensure that the value you enter is unique, it is recommended that you use the Java name space convention when providing this value. It is also useful to include the label value as part of the unique key as shown in the following examples:
	`com.<your_company_name>.nnm.trapConf.family.<family_label>`
	`com.<your_company_name>.nnm.eventConf.family.<family_label>`
	`com.<your_company_name>.nnm.inciConf.family.<family_label>`
	The maximum length is 80 alpha-numeric characters, periods allowed, no spaces allowed.

Specify the Incident Severity (Management Events)

The incident severity represents the seriousness calculated for the incident. Use the severity attribute to specify the severity that should be assigned to the incident you are configuring. Possible values are described in the following table.

Incident Severity Values

Attribute	Description
Normal	Indicates there are no known problems related to the associated object. This severity is meant to be informational. Generally, no action is needed for these incidents.
Warning	Indicates there might be a problem related to the associated object.
Minor	Indicates NNMi has detected problems related to the associated object that require further investigation.
Major	Indicates NNMi has detected problems related to the associated object to be resolved before they become critical.
Critical	Indicates NNMi has detected problems related to the associated object that require immediate attention.

See "Monitor Incidents for Problems" for more information about these severity values.

Specify Your Incident Message Format (Management Events)

When configuring an incident, specify the information you want NNMi to include in the incident's Message attribute value. You can use any combination of valid parameter strings and Custom Incident attributes to configure the Message.

Note: The incident Message limit is 1024 characters. If the returned values exceed this limit, NNMi truncates the value starting from the end of the returned text string.

"Valid Parameters for Configuring Incident Messages (Management Events)" on the next page

"Include Custom Incident Attributes in Your Message Format (Management Events)" on page 1096

Valid Parameters for Configuring Incident Messages (Management Events)

When configuring incident messages, consider using incident information as part of the message. NNMi provides the following parameter values. Use these parameters as variables when formatting an incident message.

Tip: See the Using the Incident Form for more information about the parameter values.

Note: NNMi stores varbind values as custom incident attributes (CIAs).

Tip: If a value is not stored for a parameter, it is returned as “null”.

See "Specify Your Incident Message Format (Management Events)" on the previous page for more information about configuring messages.

Parameter strings are available for the following:

Note: See the following tables to view the valid parameters for incidents generated from Custom Polled Instances: Parameter Strings for all Incidents (Attributes from an Incident form), Parameter Strings for Node Source Objects (Attributes from a Node form), and the Parameter Strings for all Incidents (Attributes not Visible from any form).

- Parameter strings for all incidents (Incident form attributes) (Click here for a list of choices.)

Parameter Strings for all Incidents (Incident form attributes)

Parameter String	Description
$category, $cat	Value of the Category attribute in the Incident form.
$count, $cnt	Value representing the number of Custom Incident Attributes that appear in the Incident form.
$family, $fam	Value from the Family attribute in the Incident form.
$firstOccurrenceTime, $fot	Value from the First Occurrence Time attribute in the incident form.
$lastOccurrenceTime, $lot	Value from the Last Occurrence Time attribute in the incident form.
$lifecycleState, $lcs	Value from the Lifecycle State attribute in the Incident form.
$name	Value of the Name attribute from the incident configuration.
$nature, $nat	Value from the Nature attribute in the Incident form.
$origin, $ori	Value from the Origin attribute in the Incident form.
$originOccurrenceTime, $oot	Value from the Origin Occurrence Time attribute in the incident form.
$priority, $pri	Value from the Priority attribute in the Incident form.
$sev, $severity	Value of the Severity attribute of the Incident form.

- Parameter Strings for Node Source Objects (Node form attributes) (Click here for a list of choices.)

Parameter Strings for Node Source Objects (Node form attributes)

Parameter String	Description
$managementAddress, $mga	Value from the Management Address attribute of the incident's source Node's form or SNMP Agent form.
$otherSideOfConnectionManagementAddress, $oma	If the incident's Source Node is part of a Layer 2 Connection, this attribute is the value of the Management Address of a node on the other side of the Layer 2 Connection.
$sourceNodeLongName, $sln	The fully-qualified DNS name as displayed in the Hostname attribute of the incident's source Node's form.
$sourceNodeName, $snn	Value from the Name attribute of the incident's source Node's form.

Parameter Strings for Node Source Objects (Node form attributes) , continued

Parameter String	Description
$sysContact, $sct	Value from the System Contact attribute of the incident's source Node form: General tab.
$sysLocation, $slc	Value from the System Location attribute of the incident's source Node form: General tab.

- Parameter Strings for Interface Source Objects (Interface form attributes) (Click here for a list of choices.)

Parameter Strings for Interface Source Objects (Interface form attributes)

Parameter String	Description
$ifAlias, $ifa	Value from the IfAlias attribute for the interface that is the incident's source object.
$ifConfigDupSetting, $icd	Configured Duplex Setting on the port associated with the interface that is the incident's source object.
$ifDesc, $idc	Value from the ifDesc attribute for the interface that is the incident's source object.
$ifIndex, $idx	Value from the ifIndex attribute for the interface that is the incident's source object.
$ifIpAddr, $iia	IP Address values associated with the interface that is the incident's source object. If multiple IPaddresses are associated with the interface, this parameter returns a comma-separated list.
$ifName, $ifn	Value from the ifName attribute for the interface that is the incident's source object.
$ifPhysAddr, $ipa	Value from the Physical Address attribute for the interface that is the incident's source object.
$ifSpeed, $isp	Value from the ifSpeed attribute for the interface that is the incident's souce object.
$ifType, $itp	Value from the ifType attribute for the interface that is the incident's souce object.

- Parameter Strings for Layer 2 Connection Source Objects (Layer 2 Connection form attributes) (Click here for a list of choices.)

Parameter Strings for Layer 2 Connection Source Objects (Layer 2 Connection form attributes)

Parameter String	Description
$otherSideOfConnectionConfigDupSetting, $ocd	If the incident's source Node is part of a Layer 2 Connection, this parameter contains the Configured Duplex Setting on the port associated with the interface on the other side of the connection.
$otherSideOfConnectionIfAlias, $oia	If the incident's Source Node is part of a Layer 2 Connection, this parameter is the value of the ifAlias of one of the interfaces on the other side of the Layer 2 Connection.
$otherSideOfConnectionIfDesc, $odc	If the incident's Source Node is part of a Layer 2 Connection, this parameter contains the ifDescr attribute value for the interface on the other side of the Layer 2 Connection.
$otherSideOfConnectionIfIndex, $odx	If the incident's Source Node is part of a Layer 2 Connection, this parameter contains the ifIndex attribute value for the interface on the other side of the connection.
$otherSideOfConnectionIfName, $ofn	If the incident's Source Node is part of a Layer 2 Connection, this parameter contains the ifName attribute value for the interface on the other side of the connection.

- Parameter strings for VLAN Source Objects (VLAN form attributes) (Click here for a list of choices.)

Parameter Strings for VLAN Source Objects (VLAN form attributes)

Parameter String	Description
$impVlanIds, $ivi	Value from the VLAN Id attribute associated with the interface that is the incident's source object. To access this information from an interface form, navigate to the VLAN Port tab and open the form for the VLAN of interest. If the interface is part of more than one VLAN, this parameter returns a comma-separated list.
$impVlanNames, $ivn	Value from the VLAN Name attribute associated with the interface that is the incident's source object. To access this information from an interface form, navigate to the VLAN Ports tab of the Interface form. If the interface is part of more than one VLAN, this parameter returns a comma-separated list.

- Parameter Strings for all incidents (Additional information that is not visible in any form) (Click here for a list of choices.)

Parameter Strings for all Incidents (Attributes not visible in any form)

Parameter String	Description
$firstOccurrenceTimeMs, $fms	Value from the First Occurrence Time attribute in the incident form, converted to millseconds (measured since January 1, 1970, 00:00:00 GMT - Greenwich Mean Time).
$lastOccurrenceTimeMs, $lms	Value from the Last Occurrence Time attribute in the incident form, converted to millseconds (measured since January 1, 1970, 00:00:00 GMT - Greenwich Mean Time).
$oid	Value of the unique object identifier (oid) for the incident configuration that originated from either an SNMP trap, Management Event, or Remote NNM 6.x or 7.x event.
$otherSideOfConnection, $osc	If the incident's Source Node is part of a Layer 2 Connection, this attribute is the following combination of values for the node and one of its interfaces on the other side of the Layer 2 Connection: The fully-qualified DNS name of the node appended with the interface Name in the following format: <*fully-qualified DNS name*>[*interface_name*]
$originOccurrenceTimeMs $oms	Value from the Origin Occurrence Time attribute in the incident form, converted to millseconds (measured since January 1, 1970, 00:00:00 GMT - Greenwich Mean Time).
$sourceNodeUuid, $snu	Universally Unique Object Identifier attribute value of the source node object for the incident (unique across all databases). This identifier distinguishes the source node object instance from all other node objects.
$sourceObjectClass, $soc	Value of the object class for the object you want to include. Use this parameter to request more details of a class of objects through a web service. Examples of object classes include: `com.hp.ov.nms.model.core.Interface` and `com.hp.ov.nms.model.snmp.SnmpAgent`.
$sourceObjectName, $son	Value from the Name attribute of the source object. For example, an interface object is named according to the MIB ifName. Each ifName varies according to the vendor's conventions. Using the name `4/1` as an example, `4` represents `the board number and 1` represents the port number.
$sourceObjectUuid, $sou	Universally Unique Object Identifier attribute value of the source object for the incident (unique across all databases). This identifier distinguishes the source object instance from all other similar object instances.
$uuid	Universally Unique Object Identifier attribute value of the

Parameter Strings for all Incidents (Attributes not visible in any form), continued

Parameter String	Description
	incident (unique across all databases). This identifier distinguishes the incident object instance from all other incident objects.

- Information established in Custom Incident Attributes (Click here for a list of choices.)

Parameter Strings for Attributes Established in Custom Incident Attributes

Parameter String	Description
$<position_number>	Value of the custom incident attribute (CIA) position number for any CIA that originated from a varbind or was added by NNMi. For example, to indicate you want to use the varbind in position 1, enter: `$1` NNMi stores varbind values as Custom Incident Attributes. If you know the varbind position number, use this parameter.
$<CIA_name>	Value of the name that is used for the custom incident attribute. For example, `$mycompany.mycia`. NNMi provides CIA values for configuring Management Events. See Custom Incident Attributes Provided by NNMi for more information about custom incident attributes.
$<CIA_oid>	Value of the object identifier for any custom incident attribute that originated as a varbind. For example, `$.1.3.6.1.6.3.1.1.5.1`. Use this parameter when you are not certain of a custom incident attribute (varbind) position number.
$*	Used to indicate you want all of the custom incident attribute values originating as varbinds, to be passed to the action configuration. Each varbind is returned in the following format: `$<CIA_name>:<CIA_value>` in which the custom incident attribute name appears followed by the custom incident attribute value.

- Functions to generate values (Click here for a list of choices.)

  The function described in the following table replaces the specified numeric value with the associated text value stored in the CIA.

  Note: The associated MIB must have been loaded using the nnmloadmib.ovpl command.

Functions to Generate Values Within the Incident Message

Function	Description
$oidtext ($<position_number>)	A <*position_number*> argument specifies the numeric value of the custom incident attribute (CIA) position number for any CIA that originated from a varbind or was added by NNMi. For example, `$oidtext($2)`. **Note**: The position number you enter must represent a CIA that contains an Object Identifier (OID) value. NNMi returns the textual value of the OID for the CIA specified.

Functions to Generate Values Within the Incident Message, continued

Function	Description
	Note the following: ▪ If the MIB is not loaded, NNMi returns the numeric OID value. ▪ If the OID has a MIB instance, the number representing the MIB instance is appended to the textual OID value.
$oidtext ($<CIA_oid>)	The <*CIA_oid*> argument specifies the Object Identifier (OID) for any custom incident attribute that originated as a varbind. For example, `$oidtext ($.1.3.6.1.6.3.1.1.5.1.)` Use this argument to the $oidtext() function when you are not certain of a custom incident attribute (varbind) position number. NNMi replaces the numeric value with the textual value of the OID you specify. Note the following: ▪ If the MIB is not loaded, NNMi returns the numeric OID value. ▪ If the OID has a MIB instance, the number representing the MIB instance is appended to the textual OID value.
$text ($<position_number>)	The <*position_number*> argument specifies the numeric value of the custom incident attribute (CIA) position number for any CIA that originated from a varbind or was added by NNMi. For example, to indicate you want to use the varbind in position 1, enter: `$1.` NNMi replaces the numeric value with the text value stored in the CIA. **Note**: If a text value is not available, NNMi returns the numeric value.
$text ($<CIA_oid>)	The <CIA_oid> argument specifies the object identifier for any custom incident attribute that originated as a varbind. For example, `$.1.3.6.1.6.3.1.1.5.1.` Use this argument to the $text function when you are not certain of a custom incident attribute (varbind) position number. NNMi replaces the numeric value with the text value stored in the CIA. **Note**: If a text value is not available, NNMi returns the numeric value.

Include Custom Incident Attributes in Your Message Format (Management Events)

NNMi includes two categories of CIAs:

- SNMP trap varbinds identified by the Abstract Syntax Notation value (ASN.1). NNMi turns varbinds into CIAs and maintains each varbind's position number. See "Load SNMP Trap Incident Configurations" on page 771.
- Custom incident attributes provided by NNMi. See "Custom Incident Attributes Provided by

NNMi (Information for Administrators)" on page 647.

You cannot create Custom Incident Attributes.

You can use CIAs in your message format to extend the amount of information presented. To determine which CIAs are available for any particular incident type, open an Incident view, locate the incident and open the Incident form. Navigate to the **Custom Attributes** tab. A complete list of available CIAs (for that incident type) appears in the table.

To include a CIA in your message format, type the dollar-sign character (`$`) plus any of the following:

- Varbind position number or asterisk (*) to include all varbind values
- Name of the CIA
- Object identifier (oid) of the CIA (useful when the varbind position number is not consistent among vendors)

Note: A single incident cannot include two CIAs with the same name. However, two incidents can contain CIAs having the same names and values.

The following table presents some example formats with the subsequent output.

Example Incident Message Formats

Example Message Format	Output in Incident View
Possible trouble with $3	`Possible trouble with` <varbind 3>
Possible trouble with $11	`Possible trouble with` <varbind 11>
Possible trouble with $77 (where the varbind position 77 does not exist)	`Possible trouble with <Invalid or unknown cia> 77`
Possible trouble with $*	`Possible trouble with` <cia1_name: cia_value>, <cia2_name; cia_value>,< cia*n*_name: cia_value>
Possible trouble with $3x	`Possible trouble with` <varbind 3>`x`
Possible trouble with $1.2.3.4.5	`Possible trouble with` <value of the CIA with oid of 1.2.3.4.5>
Possible trouble with $mycia.mycompany	`Possible trouble with` <value of the CIA with name of mycia.mycompany>

Tip: NNMi provides an error message when a CIA cannot be found. For example, if you enter an unavailable varbind position, name, or object identifier (oid), NNMi returns an "Invalid or unknown cia" error message.

Specify a Description for Your Incident Configuration (Management Events)

NNMi provides the Description attribute to help you further identify the current incident configuration.

Description

Use the description field to provide additional information that you would like to store about the current incident configuration. This description applies only to the configuration entry.

Type a maximum of 1024 characters. Alpha-numeric, spaces, and special characters (~ ! @ # $ % ^ & * ( ) _+ -) are permitted.

Configure Interface Settings for a Management Event Incident

Note: Interface Settings override any other Suppression, Enrichment, Dampen, or Actions settings for this incident, including those configured on the Node Settings tab.

NNMi enables you to apply an incident configuration to a Source Object based on the Source Object's participation in an Interface Group. If the Source Object is not a member of the Interface Group specified, the incident is neither displayed nor stored in the NNMi database

Tip: See "Create Interface Groups" on page 321 for more information about Interface Groups.

For information about each Interface Settings tab:

For information about each Management Events tab:

To apply an incident configuration to a Source Object based on the Source Object's Interface Group:

1. Navigate to the **Management Event Configuration** form:
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Expand the **Incidents** folder.
   c. Select **Management Event Configurations** .
   d. Do one of the following:
      i. To create an incident configuration, click the New icon, and continue.
      ii. To edit an incident configuration, select a row, click the Open icon, and continue.
      iii. To delete an incident configuration, select a row, and click the Delete icon.
2. Select the **Interface Settings** tab.
3. Do one of the following:
   a. To create a new configuration, click the New icon.
   b. To edit an existing configuration, double-click the row representing the configuration you want to edit.
4. Configure the desired Interface Settings (see table).
5. Configure any Suppression, Dampen, or Enrichment settings for this Interface Group.
6. Click **Save and Close** to save your changes and return to the previous form.

Interface Group Attributes

Name	Description
Interface Group	Click the Lookup icon and select Quick Find to select the Interface Group you want to use. See "Use the Quick Find Window" on page 41 for more information about using Quick Find.
Ordering	Determines the priority order for those interfaces that appear in multiple Interface Groups. The lower the number, the higher the priority. For example, **1** is the highest priority. If an interface is in multiple Interface Groups and more than one of those Interface Groups have been specified in an incident configuration, only the incident configuration with the highest priority will be applied to the interface.
Enable	Use this attribute to temporarily disable an incident's configuration settings: **Enable** □ = Temporarily disable the selected configuration. **Enable** ☑ = Enable the selected configuration.

Related Topics

"Configure Node Settings for a Management Event Incident" on page 1135

Configure Incident Suppression Settings for an Interface Group (Management Events)

Note: Interface Settings override any other Suppression settings for this incident, including those from the Node Settings tab.

NNMi enables you to suppress a specified incident configuration based on the Source Object's participation in an Interface Group.

Note: You can also suppress the incident configuration based on the Source Node's participation in a Node Group. See "Configure Incident Suppression Settings for a Node Group (Management Events)" on page 1136 for more information.

Tip: See "Create Interface Groups" on page 321 for more information about Interface Groups.

For information about each Interface Settings tab:

To suppress an incident configuration based on an Interface Group:

1. Navigate to the **Management Event Configuration** form:
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Expand the **Incidents** folder.
   c. Select **Management Event Configurations** .
   d. Do one of the following:
      i. To create an incident configuration, click the * New icon, and continue.

  ii. To edit an incident configuration, select a row, click the Open icon, and continue.

  iii. To delete an incident configuration, select a row, and click the Delete icon.

2. Select the **Interface Settings** tab.
3. Do one of the following:
   a. To create a new configuration, click the New icon.

   b. To edit a configuration, double-click the row representing the configuration you want to edit.
4. Make sure you configure the basic Interface Setting behavior. See "Configure Interface Settings for a Management Event Incident" on page 1098 for more information.
5. Select the **Suppression** tab.
6. Configure the desired Suppression behavior (see table).
7. Click **Save and Close** to save your changes and return to the previous form.

Interface Settings Suppression Attributes

Name	Description
Enable	Use this attribute to temporarily disable an incident's suppression settings for the specified Interface Group: **Enable** □ = Temporarily disable the selected configuration. **Enable** ☑ = Enable the selected configuration.
Payload Filter	The Payload Filter Editor enables you to create expressions that further refine the filters used to select the incidents to be suppressed, enriched, or dampened. Make sure to design any complex Payload Filters offline as a Boolean expression first. This method can help to minimize errors when entering your expressions using the Payload Filter editor. When creating a Payload Filter, note the following: • Payload Filter expressions for the `like` and `not like` operators use the syntax defined for java regular expressions (java.util.regex Pattern class). • You must use a `ciaName` that already exists in the trap or event you are configuring. • Each set of expressions associated with a Boolean Operator is treated as if it were enclosed in parentheses and evaluated together. • View the expression displayed under **Filter String** to see the logic of the expression as it is created. • The `AND` and `OR` Boolean Operators must contain at least two expressions as shown in the example below. The following example filters incidents on voltage state: `AND`

Interface Settings Suppression Attributes , continued

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td></td><td>
<code>ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7</code>
<code>ciaValue = 5</code>
NNMi evaluates the expression above as follows:
<code>(ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7 AND ciaValue = 5)</code>
NNMi finds all incidents with a varbind <code>.1.3.6.1.4.1.9.9.13.1.2.1.7</code> value of <b>5</b>.
• The placement of your cursor and the subsequent text that is selected is important when performing operations using the Payload Filter Editor. For example, you append to, replace, or change the indentation of the expression that is selected.
• The placement of your cursor and the subsequent text that is selected is especially important when adding your Boolean operators.
• You can include more than one varbind in the same Payload Filter expression as shown in the following example:
<code>((ciaName like \Q.1.3.6.1.4.1.9.9\E.* AND ciaValue = 25) AND (ciaName like \Q.1.3.6.1.2.1.2.2.1.1.3\E.* AND ciaValue = 3))</code>
In this example, a given trap must meet each of the following criteria:
▪ Contain a varbind whose Object Identifier (OID) matches the regular expression <code>\Q.1.3.6.1.4.1.9.9\E.*</code> and has a value of <code>25</code>.
▪ Contain a varbind whose OID matches the regular expression <code>\Q.1.3.6.1.2.1.2.2.1.1.3\E.*</code> and has a value of <code>3</code>.
<b>Payload Filter Editor Components</b>
<table>
<tr><th>Attribute</th><th>Description</th></tr>
<tr><td>Attribute</td><td>The attribute name on which NNMi searches. Filterable attributes include the following:
• ciaName
• ciaValue</td></tr>
<tr><td>Operator</td><td>Valid operators are described below.
• <b>=</b> Finds all values equal to the value specified. Click here for an example.
Example: <code>ciaName=.1.3.6.1.4.1.9.9.13.1.2.1.7</code> matches any incident that contains a varbind with the name value of <b>.1.3.6.1.4.1.9.9.13.1.2.1.7</b>.
• <b>!=</b> Finds all values not equal to the value specified. Click here for an example.</td></tr>
</table>
</td></tr>
</table>

Interface Settings Suppression Attributes , continued

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td></td><td>

Payload Filter Editor Components, continued

<table>
<tr><th>Attribute</th><th>Description</th></tr>
<tr><td></td><td>

Example: `ciaName! = .1.3.6.1.4.1.9.9.13.1.2.1.7` matches any incident that contains a varbind with a name value other than **1.3.6.1.4.1.9.9.13.1.2.1.7**.

- **<** Finds all values less than the value specified. Click here for an example.

  Example: `ciaValue < 6` matches any incident that contains a varbind with a value less than **6**.

- **<=** Finds all values less than or equal to the value specified. Click here for an example.

  Example: `ciaValue <= 6` matches any incident that contains a varbind with a value less than or equal to **6**.

- **>** Finds all values greater than the value specified. Click here for an example.

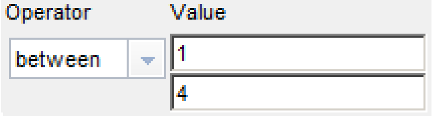
  Example: `ciaValue > 4` matches any incident that contains a varbind with a value greater than **4**.

- **>=** Finds all values greater than or equal to the value specified. Click here for an example.

  Example: `ciaValue >= 4` matches any incident that contains a varbind with values greater than or equal to **4**.

- **between** Finds all values equal to and between the two values specified. Click here for an example.

  Example: `ciaValue between`

  Operator: between
  Value: 1
  4

  matches any incident that contains a varbind value equal to or greater than **1** and equal to or less than **4**.

  Note: As shown in the example, each value must be entered on a separate line.

- **in** Finds any match to at least one value in a list of values. Click here for an example.

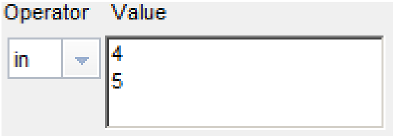
  Example:

  `ciaValue in`

</td></tr>
</table>

</td></tr>
</table>

Interface Settings Suppression Attributes , continued

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td></td><td>
<b>Payload Filter Editor Components, continued</b>
<table>
<tr><th>Attribute</th><th>Description</th></tr>
<tr><td></td><td>

Operator Value
in
4
5

matches any incident with a varbind value of either <b>4</b> or <b>5</b>.

<b>Note</b>: As shown in the example, each value must be entered on a separate line.

NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example (<b>4, 5</b>). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line.

• <b>is not null</b> Finds all non-blank values. Click here for an example.
Example: <code>ciaValue is not null</code> matches any incident with a varbind that contains a value.

• <b>is null</b> Finds all blank values. Click here for an example.
Example: <code>ciaValue is null</code> matches any incident with a varbind that does not contain a value.

• <b>like</b> Finds matches using the syntax defined for java regular expressions. See the Pattern (Java Platform SE6) API documentation at : <code>http://download.oracle.com/javase/6/docs/api/java/util/regex/Pattern.html</code> for more information. Click here for more information.

The period asterisk (.*) characters mean <i>any number of characters of any type at this location</i>.

The period (.) character means <i>any single character of any type at this location</i>.

<b>Note</b>: To include literal string values in the Value attribute, enclose the string value in \Q&lt;<i>literal_value</i>&gt;\E as shown in the Examples listed below.

Example:

<code>ciaName like \Q.1.3.6.1.4.1.9.9\E.*</code> matches any incident that contains a varbind name value that begins with 1.3.6.1.4.1.9.9 and (optionally) ends with any number of characters.

<code>ciaValue like .*Chicago.*</code> finds all traps or events that contain a
</td></tr>
</table>
</td></tr>
</table>

Interface Settings Suppression Attributes , continued

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td></td><td>

Payload Filter Editor Components, continued

<table>
<tr><th>Attribute</th><th>Description</th></tr>
<tr><td></td><td>

varbind value that includes the string **Chicago**.

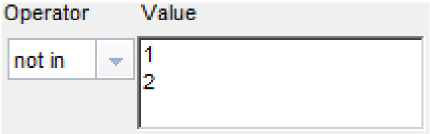
- **not between** Finds all values except those between the two values specified. Click here for an example.

  Example: `ciaValue not between 5 8` matches an incident that contains a varbind with the values less than **5** or greater than **8** .

- **not in** Finds all values except those included in the list of values. Click here for an example.

  Example:

  `ciaValue not in`

  Operator: not in
  Value:
  1
  2

  matches any incident that contains a varbind with values other than **1** and **2**.

  Note: As shown in the example, each value must be entered on a separate line.

  NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example, (**1, 2**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line.

- **not like** Finds all that do not have the values specified using the syntax defined for Java regular expressions. See the Pattern (Java Platform SE6) API documentation at: `http://download.oracle.com/javase/6/docs/api/java/util/regex/Pattern.html` for more information. Click here for an example.

  The period asterisk (.*) characters mean *any number of characters of any type at this location*.

  The period (.) character means *any single character of any type at this location*.

  Note: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below.

</td></tr>
</table>

</td></tr>
</table>

Interface Settings Suppression Attributes , continued

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td></td><td>

Payload Filter Editor Components, continued

<table>
<tr><th>Attribute</th><th>Description</th></tr>
<tr><td></td><td>

Example:

`ciaName not like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that does not begin with **.1.3.6.1.4.1.9.9** and (optionally) ends with any number of characters.

`ciaValue not like .*Chicago.*` finds all traps or events that do not contain a varbind value that includes the string **Chicago**.

</td></tr>
<tr><td>Value</td><td>

The value for which you want NNMi to search.

Note the following:

- The values you enter are case sensitive.
- NNMi displays a variable number of value fields depending on the Operator selected. For example, the `between` Operator causes two value fields to be displayed.
- The `between, in` and `not in` operators require that each value be entered on a separate line.

</td></tr>
</table>

Payload Filter Editor Buttons

<table>
<tr><th>Button</th><th>Description</th></tr>
<tr><td>Append</td><td>Appends the current expression (Attribute, Operator,and Value) to the selected expression already included in the filter string.</td></tr>
<tr><td>Insert</td><td>Inserts the current expression (Attribute, Operator,and Value) in front of the cursor location within the Filter String.</td></tr>
<tr><td>Replace</td><td>Replaces the selected expression with the expression displayed in the Attribute, Operator, and Value fields.</td></tr>
<tr><td>AND</td><td>

Inserts the AND Boolean Operator in the selected cursor location.

Note: View the expression displayed under **Filter String** to see the logic of the expression as it is created.

</td></tr>
<tr><td>OR</td><td>

Inserts the OR Boolean Operator in the current cursor location.

Note: View the expression displayed under **Filter String** to see the logic of the expression as it is created.

</td></tr>
<tr><td>NOT</td><td>Can be used in any part of the Filter String to specify that NNMi should</td></tr>
</table>

</td></tr>
</table>

Interface Settings Suppression Attributes , continued

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td></td><td>
<b>Payload Filter Editor Buttons, continued</b>
<table>
<tr><th>Button</th><th>Description</th></tr>
<tr><td></td><td>exclude interfaces with values that pass the expression that immediately follows the NOT.
For example, when evaluating the following expression, NNMi includes interfaces with (interface description) ifDesc containing <b>VLAN</b>, and excludes any Interfaces that have <b>VLAN10</b> for the (interface name) ifName value:
(ifDesc like VLAN AND NOT (ifName=VLAN10))
<b>Note</b>: View the expression displayed under Filter String to see the logic of the expression as it is created .</td></tr>
<tr><td>EXISTS</td><td>Used for filters that include Capabilities or Custom Attribute names and values in the Filer String.
Indicates that you want NNMi to consider interfaces that do not have any Capabilities or Custom Attributes when evaluating the Filter String.
<b>Note</b>: If you include Capabilities or Custom Attribute names and values in the Filer String, but do not use <b>EXISTS</b> or <b>NOT EXISTS</b>, NNMi excludes from its search interfaces that do not include Capabilities or Custom Attributes.
For example, when evaluating the following Filter String, NNMi includes interfaces with (interface description) ifDesc containing <b>VLAN</b>, as well as any Interfaces Custom Attribute Role value is <b>LAN Connection to Oracle Server</b>:
(ifDesc like VLAN OR EXISTS((customAttrName=Role AND customAttrValue=LAN Connection to Oracle Server)))
<b>Note</b>: View the expression displayed under Filter String to see the logic of the expression as it is created.</td></tr>
<tr><td>NOT EXISTS</td><td>Used for filters that include Capabilities or Custom Attribute names and values in the Filer String. Indicates that you want NNMi to consider interfaces that do not have any Capabilities or Custom Attributes when evaluating the Filter String, but exclude the interfaces that match the expression that follows the <b>NOT EXISTS</b>.
<b>Note</b>: If you include Capabilities or Custom Attribute names and values in the Filer String, but do not use <b>EXISTS</b> or <b>NOT EXISTS</b>, NNMi excludes from its search interfaces that do not</td></tr>
</table>
</td></tr>
</table>

Interface Settings Suppression Attributes , continued

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td></td><td>

Payload Filter Editor Buttons, continued

<table>
<tr><th>Button</th><th>Description</th></tr>
<tr><td></td><td>include Capabilities or Custom Attributes.
For example, when evaluating the following expression, NNMi includes interfaces with (interface description) `ifDesc` containing **VLAN**, and excludes any Interfaces that have the Custom Attribute Role and that Role value is **LAN Connection to Oracle Server**:
`(ifDesc like VLAN OR NOT EXISTS ((customAttrName=Role AND customAttrValue=LAN Connection to Oracle Server)))`
Note: View the expression displayed under Filter String to see the logic of the expression as it is created.</td></tr>
<tr><td>Delete</td><td>Deletes the selected expression.
Note: If the Boolean Operator is selected, the Payload Filter Editor deletes all expressions associated with the Boolean Operator.</td></tr>
</table>

</td></tr>
</table>

Configure Incident Enrichment Settings for an Interface Group (Management Events)

Note: Interface Settings override any other Enrichment settings for this incident, including those from the Node Settings tab.

NNMi enables you to fine tune and enhance a specified incident configuration based on the Source Object's participation in an Interface Group. The types of items you can fine tune and enhance for a selected incident configuration, include:

- Category
- Family
- Severity
- Priority
- Correlation Nature
- Message
- Assigned To

Note: You can also enhance the incident configuration based on the Source Node's participation in a Node Group. See "Configure Incident Enrichment Settings for a Node Group (Management Events)" on page 1143 for more information.

Tip: See Create Interface Groups for more information about Interface Groups.

For information about each Interface Settings tab:

For information about each Enrichment tab:

To enrich an incident configuration based on an Interface Group:

1. Navigate to the **Management Event Configuration** form:
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Expand the **Incidents** folder.
   c. Select **Management Event Configurations** .
   d. Do one of the following:
      i. To create an incident configuration, click the ✱ New icon, and continue.
      ii. To edit an incident configuration, select a row, click the 📂 Open icon, and continue.
      iii. To delete an incident configuration, select a row, and click the ✖ Delete icon.
2. Select the **Interface Settings** tab.
3. Do one of the following:
   a. To create a new configuration, click the ✱ New icon.
   b. To edit an existing configuration, double-click the row representing the configuration you want to edit.
4. Make sure you configure the basic Interface Setting behavior. See "Configure Interface Settings for a Management Event Incident" on page 1098 for more information.
5. Select the **Enrichment** tab.
6. Do one of the following:
   a. To create an Enrichment configuration, click the ✱ New icon and continue.
   b. To edit an Enrichment configuration, select a row, click the 📂 Open icon, and continue.
   c. To delete an Enrichment configuration, select a row and click the ✖ Delete icon.
7. Configure the desired Enrichment behavior (see the "Enrich Configuration Attributes" table)
8. Click 💾 **Save and Close** to save your changes and return to the previous form.

Interface Settings Enrichment Configuration Attributes

Name	Description
Category	Use the Category attribute to customize the category for this incident configuration. Possible values include: • Accounting • Application Status

Interface Settings Enrichment Configuration Attributes , continued

Name	Description
	• Configuration • Fault • Performance • Security • Status See "Specify Category and Family Attribute Values for Organizing Your Incidents (SNMP Trap Incident)" on page 790 for more information.
Family	Use the Family attribute to customize the Family for this incident configuration. Select from the drop-down list or create a new value. For example, some of the values provided by NNMi include: • Address • Aggregated Port (Interfaces using **Link Aggregation**[1] protocol. See Interface Form: Link Aggregation tab.) • Card • Connection • Correlation • Interface • Node
Severity	The incident Severity represents the seriousness calculated for the incident. Use the Severity attribute to specify the severity that should be assigned to the incident you are configuring. Possible values are described below: **Normal** - Indicates there are no known problems related to the associated object. This Severity is meant to be informational. Generally, no action is needed for these incidents. **Warning** - Indicates there might be a problem related to the associated object. **Minor** - Indicates NNMi has detected problems related to the associated object that require further investigation. **Major** - Indicates NNMi has detected problems related to the associated object to

[1] Protocols used on Switches to configure multiple Interfaces (Aggregation Member Interfaces) to function as if they were one (an Aggregator Interface). When two Aggregator Interfaces establish a connection, that connection is an Aggregator Layer 2 Connection. The Aggregator Layer 2 Connection appears on Layer 2 Neighbor View maps as a thick line with an Interface icon at each end (representing the Aggregator Interface). When you double-click the thick line, it converts into multiple thin lines representing the participating Aggregation Member Layer 2 Connections with their Aggregation Member Interfaces at each end of the lines.

Interface Settings Enrichment Configuration Attributes , continued

Name	Description
	be resolved before they become critical. **Critical** - Indicates NNMi has detected problems related to the associated object that require immediate attention.
Priority	Used to communicate the urgency of resolving the selected incident. You control this value. NNMi sets this value to null by default. The lower the number the higher the priority. Possible values are: 5 **None** 4 **Low** 3 **Medium** 2 **High** 1 **Top** **Note**: The icons are displayed only in table views.
Correlation Nature	Use the Correlation Nature to customize the Correlation Nature for this incident configuration. Possible values include: • Info • None • Root Cause • Secondary Root Cause • Symptom • Stream Correlation • Service Impact • Dedup Stream Correlation • Rate Stream Correlation See Incident Form: General Tab for more information.
Message Format	When configuring an incident, specify how the incident message appears in the incident view. The string you specify in the Message Format attribute is visible in an incident view. **Note**: The incident message limit is 1024 characters. If you exceed this limit, NNMi truncates the value starting from the right. You can use any combination of default and custom attributes: "Valid Parameters for Configuring Incident Messages (Management Events)" on

Interface Settings Enrichment Configuration Attributes , continued

Name	Description
	page 1090 "Include Custom Incident Attributes in Your Message Format (Management Events)" on page 1096
Assigned To	Use to specify the owner of any incident generated for this incident configuration. Click the Lookup icon and select Quick Find to select a valid user name. **Note**: You can also begin to type a valid user name and use the auto-complete feature to select the user name of interest.
Description	Use the Description attribute to provide additional information that you want to note about the current enhancement configuration. This description applies only to the enhancement configuration and does not appear when NNMi displays any associated incident. Type a maximum of 1024 characters. Alpha-numeric and special characters (~ ! @ # $ % ^ & * ( ) _+ -) are permitted.

Configure Custom Incident Attributes to Enrich an Incident Configuration (Interface Settings) (Management Events)

The Custom Incident Attributes (CIAs) tab enables you to specify additional CIAs to be saved with an incoming incident. The persisted data might then be used as an argument to an action for the incident.

When creating a CIA for an incident configuration, you can specify any of the following values:

- Custom Attribute on the source node
- Custom Attribute on the interface (source object)

You also specify the Custom Incident Attribute name that will store this information.

For information about each Enrichment tab:

To create a Custom Incident Attribute to enrich an incident configuration:

1. Navigate to the **Management Event Configuration** form:
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Expand the **Incidents** folder.
   c. Select **Management Event Configurations** .
   d. Do one of the following:
      i. To create an incident configuration, click the ✱ New icon, and continue.

   ii. To edit an incident configuration, select a row, click the Open icon, and continue.

   iii. To delete an incident configuration, select a row, and click the Delete icon.

2. Select **Interface Settings**.
3. Do one of the following:
   a. To create a new configuration, click the New icon.

   b. To edit an existing configuration, double-click the row representing the configuration you want to edit.
4. Make sure you configure the basic Interface Setting behavior. See "Configure Interface Settings for a Management Event Incident" on page 1098 for more information.
5. Select the **Enrichment** tab.
6. Do one of the following:

   a. To create an Enrichment configuration, click the New icon, and continue.

   b. To edit an Enrichment configuration,select a row, click the Open icon, and continue.

   c. To delete an Enrichment configuration, select a row and click the Delete icon.
7. Make sure you configure the Enrichment settings. See "Configure Incident Enrichment Settings for an Interface Group (Management Events)" on page 1107 for more information.
8. Navigate to the **Custom Incident Attributes** tab.
9. Do one of the following:

   a. To create a Custom Incident Attribute, click the New icon, and continue.

   b. To edit a Custom Incident Attribute, select a row, click the Open icon, and continue.

   c. To delete a Custom Incident Attribute, select a row and click the Delete icon.
10. Configure the Custom Incident Attribute (see the "Custom Incident Attribute" table).
11. Click **Save and Close** to save your changes and return to the previous form.

Custom Incident Attribute

Name	Description
Custom Incident Attribute Name	Name used to identify the Custom Incident Attribute you are configuring. The name limit is 255 characters. Alpha-numeric characters are permitted. No spaces or special characters (~ ! @ # $ % ^ & * ( ) _+ -) are permitted.
Type	Specifies whether you are using a Custom Attribute on a node or a Custom Attribute on an interface. Possible values are: • Node Custom Attribute • Interface Custom Attribute
Custom Attribute	Used to determine the value to be assigned to the Custom Incident Attribute you are configuring. Enter either of the following:

Custom Incident Attribute , continued

Name	Description
Name	• Name of the Custom Attribute on the source node • Name of the Custom Attribute on the interface (source object)

Configure a Payload Filter to Enrich an Incident Configuration (Interface Settings) (Management Events)

The Payload Filter Editor enables you to create expressions that further refine the filters used to select the incidents to be enriched. Make sure to design any complex Payload Filters offline as a Boolean expression first. This method can help to minimize errors when entering your expressions using the Payload Filter editor.

To create a Payload Filter expression:

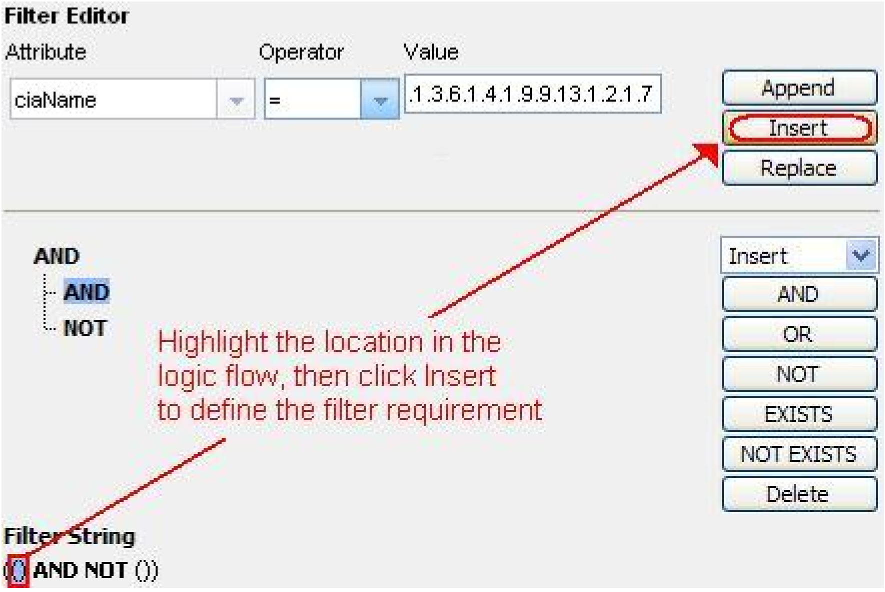
1. Navigate to the **Management Event Configuration** form:
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Expand the **Incidents** folder.
   c. Select **Management Event Configurations** .
   d. Do one of the following:
      i. To create an incident configuration, click the ✱ New icon, and continue.
      ii. To edit an incident configuration, select a row, click the 🗁 Open icon, and continue.
      iii. To delete an incident configuration, select a row, and click the ✖ Delete icon.
2. Select the **Interface Settings** tab.
3. Do one of the following:
   a. To create a new configuration, click the ✱ New icon.
   b. To edit an existing configuration, double-click the row representing the configuration you want to edit.
4. Make sure you configure the basic Interface Setting behavior. See "Configure Interface Settings for a Management Event Incident" on page 1098 for more information.
5. Select the **Enrichment** tab.
6. Do one of the following:
   a. To create an Enrichment configuration, click the ✱ New icon, and continue.
   b. To edit an Enrichment configuration, select a row, click the 🗁 Open icon, and continue.
   c. To delete an Enrichment configuration, select a row and click the ✖ Delete icon.
7. Make sure you configure the Enrichment settings. See "Configure Incident Enrichment Settings for an Interface Group (Management Events)" on page 1107 for more information.

8. Select the **Payload Filter** tab.
9. Define your Payload Filter (see table). Also see "Guidelines for Creating a Payload Filter".
   a. Plan out the logic needed for your Filter String.
   b. Use the buttons on the bottom half of the Additional Filters Editor to establish the logic structure.

      For example, to establish the following structure, select **Insert**, then click **AND**, then **NOT**, and then **AND** a second time:

      ```
      (( ) AND NOT ( ))
      ```
   c. Now place your cursor in a location within the displayed Filter String, and use the top half of the filter editor to define the parameters of the highlighted filter requirement.

      For example, select a set of parentheses and use the Insert button to specify the filter requirement within those parentheses:



Filter Editor

Attribute: ciaName

Operator: =

Value: .1.3.6.1.4.1.9.9.13.1.2.1.7

AND
- AND
- NOT

Filter String

(() AND NOT ())

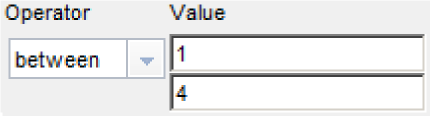
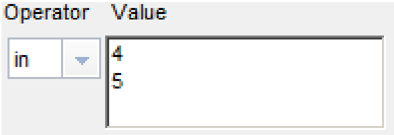
Highlight the location in the logic flow, then click Insert to define the filter requirement

10. Click **Save and Close**.
11. Click **Save and Close** to save your changes and return to the previous form.

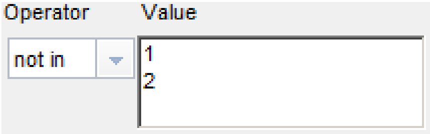
Payload Filter Editor Components

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: • ciaName • ciaValue
Operator	Valid operators are described below. • **=** Finds all values equal to the value specified. Click here for an example.

Payload Filter Editor Components, continued

Attribute	Description
	Example: `ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7` matches any incident that contains a varbind with the name value **.1.3.6.1.4.1.9.9.13.1.2.1.7**. • **!=** Finds all values not equal to the value specified. Click here for an example. Example: `ciaName != .1.3.6.1.4.1.9.9.13.1.2.1.7` matches any incident that contains a varbind with the name value other than **1.3.6.1.4.1.9.9.13.1.2.1.7**. • **<** Finds all values less than the value specified. Click here for an example. Example: `ciaValue < 6` matches any incident with a varbind value less than **6**. • **<=** Finds all values less than or equal to the value specified. Click here for an example. Example: `ciaValue <= 6` matches any incident with a varbind value less than or equal to **6**. • **>** Finds all values greater than the value specified. Click here for an example. Example: `ciaValue > 4` matches any incident with a varbind value greater than **4**. • **>=** Finds all values greater than or equal to the value specified. Click here for an example. Example: `ciaValue >= 4` matches any incident that contains a varbind with values greater than or equal to **4**. • **between** Finds all traps or events that include a varbind with a value equal to and between the two values specified. Click here for an example. Example: `ciaValue between`  Operator: between Value: 1 4 matches any incident that contains a varbind value equal to or greater than **1** and equal to or less than **4**. **Note**: As shown in the example, each value must be entered on a separate line. • **in** Finds any match to at least one value in a list of values. Click here for an example. Example: `ciaValue in`  Operator: in Value: 4 5 matches any incident with a varbind value of either **4** or **5**.

Payload Filter Editor Components, continued

Attribute	Description
	Note: As shown in the example, each value must be entered on a separate line. NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example (**4, 5**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line. - **is not null** Finds all non-blank values. Click here for an example. Example: `ciaValue is not null` matches any incident with a varbind that contains a value. - **is null** Finds all blank values. Click here for an example. Example: `ciaValue is null` matches any incident with a varbind that does not have a value. - **like** Finds matches using wildcard characters. Click here for more information about using wildcard characters. The period asterisk (.*) characters mean *any number of characters of any type at this location*. The period (.) character means *any single character of any type at this location*. **Note**: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below. Examples: `ciaName like  \Q.1.3.6.1.4.1.9.9\E.*` finds all traps or events that contain varbind names that begin with **.1.3.6.1.4.1.9.9** and (optionally) end with any number of characters. `ciaValue like .*Chicago.*` finds all traps or events that contain a varbind value that includes the string **Chicago**. - **not between** Finds all values except those between the two values specified. Click here for an example. Example: `ciaValue not between 5 8` matches an incident that contains a varbind with the values less than **5** or greater than **8**. - **not in** Finds all values except those included in the list of values. Click here for an example. Example: `ciaValue not in`  Operator: not in Value: 1 2 matches any incident that contains a varbind with values other than **1** and **2**. **Note**: As shown in the example, each value must be entered on a separate line.

Payload Filter Editor Components, continued

Attribute	Description
	NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example, (**1, 2**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line. • **not like** Finds all that do not have the values specified (using wildcard strings). Click here for an example. The period asterisk (.*) characters mean *any number of characters of any type at this location*. The period (.) character means *any single character of any type at this location*. **Note**: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below. Example: `ciaName not like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that does not begin with **.1.3.6.1.4.1.9.9**. `ciaValue not like .*Chicago.*` finds all traps or events that do not contain a varbind value that includes the string **Chicago**.
Value	The value for which you want NNMi to search. Note the following: • The values you enter are case sensitive. • NNMi displays a variable number of value fields depending on the Operator selected. For example, the `between` Operator causes two value fields to be displayed. • The `between, in` and `not in` operators require that each value be entered on a separate line.

Additional Filters Editor Buttons

Button	Description
Append	Appends the current expression (Attribute, Operator,and Value) to the selected expression already included in the Filter String.
Insert	Inserts the current expression (Attribute, Operator,and Value) in front of the cursor location within the Filter String.
Replace	Replaces the selected expression with the expression displayed in the Attribute, Operator, and Value fields.
AND	Appends, inserts, or replaces the AND Boolean Operator in the selected cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.

Additional Filters Editor Buttons, continued

Button	Description
OR	Appends, Inserts, or replaces the OR Boolean Operator in the current cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
NOT	Can be used in any part of the Filter String to specify that NNMi should exclude interfaces with values that pass the expression that immediately follows the `NOT`. For example, when evaluating the following expression, NNMi includes interfaces with (interface description) `ifDesc` containing **VLAN**, and excludes any Interfaces that have **VLAN10** for the (interface name) `ifName` value: `(ifDesc like VLAN AND NOT (ifName=VLAN10))` **Note**: View the expression displayed under Filter String to see the logic of the expression as it is created .
EXISTS	Used for filters that include Capabilities or Custom Attribute names and values in the Filer String. Indicates that you want NNMi to consider interfaces that do not have any Capabilities or Custom Attributes when evaluating the Filter String. **Note**: If you include Capabilities or Custom Attribute names and values in the Filer String, but do not use **EXISTS** or **NOT EXISTS**, NNMi excludes from its search interfaces that do not include Capabilities or Custom Attributes. For example, when evaluating the following Filter String, NNMi includes interfaces with (interface description) `ifDesc` containing **VLAN**, as well as any Interfaces Custom Attribute Role value is **LAN Connection to Oracle Server**: `(ifDesc like VLAN OR EXISTS((customAttrName=Role AND customAttrValue=LAN Connection to Oracle Server)))` **Note**: View the expression displayed under Filter String to see the logic of the expression as it is created.
NOT EXISTS	Used for filters that include Capabilities or Custom Attribute names and values in the Filer String. Indicates that you want NNMi to consider interfaces that do not have any Capabilities or Custom Attributes when evaluating the Filter String, but exclude the interfaces that match the expression that follows the **NOT EXISTS**. Note: If you include Capabilities or Custom Attribute names and values in the Filer String, but do not use **EXISTS** or **NOT EXISTS**, NNMi excludes from its search interfaces that do not include Capabilities or Custom Attributes. For example, when evaluating the following expression, NNMi includes interfaces with (interface description) `ifDesc` containing **VLAN**, and excludes any Interfaces that have the Custom Attribute Role and that Role value is **LAN Connection to Oracle Server**:

Additional Filters Editor Buttons, continued

Button	Description
	`(ifDesc like VLAN OR NOT EXISTS((customAttrName=Role AND customAttrValue=LAN Connection to Oracle Server)))` **Note**: View the expression displayed under Filter String to see the logic of the expression as it is created.
Delete	Deletes the selected expression. **Note**: If the Boolean Operator is selected, the Additional Filters Editor deletes all expressions associated with the Boolean Operator.

Configure Incident Dampening Settings for an Interface Group (Management Events)

Note: Interface Settings override any other Dampening settings for this incident, including those from the Node Settings tab.

NNMi enables you to delay the following for an incident configuration based on the Source Object's participation in an Interface Group:

- Execution of Incident Actions
- Execution of Diagnostics (*HP Network Node Manager iSPI Network Engineering Toolset Software \ NNM iSPI NET*)
- Appearance within Incident views in the NNMi Console

Note: You can also configure the Dampening settings based on the Source Node's participation in a Node Group. See "Configure Incident Dampening Settings for a Node Group (Management Events)" on page 1155 for more information.

Tip: See "Create Interface Groups" on page 321 for more information about Interface Groups.

For information about each Interface Settings tab:

When using the Dampening configuration, note the following:

- NNMi initially assigns incidents with Dampening settings configured a Lifecycle State of DAMPENED.
- After the Dampen Interval has passed, NNMi changes the Lifecycle State to REGISTERED.

See About the Incident Lifecycle for more information about Lifecycle State.

To configure the Dampening settings based on an Interface Group:

1. Navigate to the **Management Event Configuration** form:
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Expand the **Incidents** folder.
   c. Select **Management Event Configurations** .
   d. Do one of the following:

i. To create an incident configuration, click the New icon, and continue.

ii. To edit an incident configuration, select a row, click the Open icon, and continue.

iii. To delete an incident configuration, select a row, and click the Delete icon.

2. Select the **Interface Settings** tab.
3. Do one of the following:
   a. To create a new configuration, click the New icon.

   b. To edit an existing configuration, select a row, click the Open icon, and continue.
4. Make sure you configure the basic Interface Setting behavior. See "Configure Interface Settings for a Management Event Incident" on page 1098 for more information.
5. Select the **Dampening** tab.
6. Configure the desired Dampening behavior (see table).
7. Click **Save and Close** to save your changes and return to the previous form.

Interface Settings Dampening Configuration Attributes

Name	Description
Enable	Use this attribute to temporarily disable an incident's dampening settings: **Enable** □ = Temporarily disable the selected configuration. **Enable** ☑ = Enable the selected configuration.
Hour	Specifies the number of hours to be used for the dampen interval.
Minutes	Specifies the number of minutes to be used for the dampen interval. **Note:** The maximum dampen interval is 60 minutes.
Seconds	Specifies the number of seconds to be used for the dampen interval.
Payload Filter	The Payload Filter Editor enables you to create expressions that further refine the filters used to select the incidents to be suppressed, enriched, or dampened. Make sure to design any complex Payload Filters offline as a Boolean expression first. This method can help to minimize errors when entering your expressions using the Payload Filter editor. When creating a Payload Filter, note the following: • Payload Filter expressions for the `like` and `not like` operators use the syntax defined for java regular expressions (java.util.regex Pattern class). • You must use a `ciaName` that already exists in the trap or event you are configuring. • Each set of expressions associated with a Boolean Operator is treated as if it were enclosed in parentheses and evaluated together. • View the expression displayed under **Filter String** to see the logic of the expression

Interface Settings Dampening Configuration Attributes , continued

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td></td><td>

as it is created.

- The AND and OR Boolean Operators must contain at least two expressions as shown in the example below.

  The following example filters incidents on voltage state:

  ```
  AND
      ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7
      ciaValue = 5
  ```

  NNMi evaluates the expression above as follows:

  ```
  (ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7 AND ciaValue = 5)
  ```

  NNMi finds all incidents with a varbind .1.3.6.1.4.1.9.9.13.1.2.1.7 value of **5**.

- The placement of your cursor and the subsequent text that is selected is important when performing operations using the Payload Filter Editor. For example, you append to, replace, or change the indentation of the expression that is selected.

- The placement of your cursor and the subsequent text that is selected is especially important when adding your Boolean operators.

- You can include more than one varbind in the same Payload Filter expression as shown in the following example:

  ```
  ((ciaName like \Q.1.3.6.1.4.1.9.9\E.* AND ciaValue = 25) AND
  (ciaName like \Q.1.3.6.1.2.1.2.2.1.1.3\E.* AND ciaValue =
  3))
  ```

  In this example, a given trap must meet each of the following criteria:

  - Contain a varbind whose Object Identifier (OID) matches the regular expression \Q.1.3.6.1.4.1.9.9\E.* and has a value of 25.

  - Contain a varbind whose OID matches the regular expression \Q.1.3.6.1.2.1.2.2.1.1.3\E.* and has a value of 3.

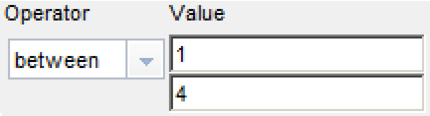
Payload Filter Editor Components

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: • ciaName • ciaValue
Operator	Valid operators are described below. • **=** Finds all values equal to the value specified. Click here for an example.

</td></tr>
</table>

Interface Settings Dampening Configuration Attributes , continued

Name	Description
	Payload Filter Editor Components, continued

Attribute	Description
	Example: `ciaName=.1.3.6.1.4.1.9.9.13.1.2.1.7` matches any incident that contains a varbind with the name value of **.1.3.6.1.4.1.9.9.13.1.2.1.7**. - **!=** Finds all values not equal to the value specified. Click here for an example. Example: `ciaName! = .1.3.6.1.4.1.9.9.13.1.2.1.7` matches any incident that contains a varbind with a name value other than **1.3.6.1.4.1.9.9.13.1.2.1.7**. - **<** Finds all values less than the value specified. Click here for an example. Example: `ciaValue < 6` matches any incident that contains a varbind with a value less than **6**. - **<=** Finds all values less than or equal to the value specified. Click here for an example. Example: `ciaValue <= 6` matches any incident that contains a varbind with a value less than or equal to **6**. - **>** Finds all values greater than the value specified. Click here for an example. Example: `ciaValue > 4` matches any incident that contains a varbind with a value greater than **4**. - **>=** Finds all values greater than or equal to the value specified. Click here for an example. Example: `ciaValue >= 4` matches any incident that contains a varbind with values greater than or equal to **4**. - **between** Finds all values equal to and between the two values specified. Click here for an example. Example: `ciaValue between` Operator: between Value: 1 4 matches any incident that contains a varbind value equal to or greater than **1** and equal to or less than **4**. **Note**: As shown in the example, each value must be entered on a separate line. - **in** Finds any match to at least one value in a list of values. Click here for an example.

Interface Settings Dampening Configuration Attributes , continued

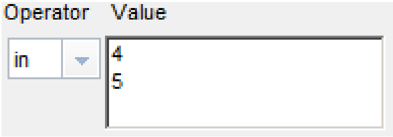
Name	Description

Payload Filter Editor Components, continued

Attribute	Description

Example:

`ciaValue in`



Operator: in
Value:
4
5

matches any incident with a varbind value of either **4** or **5**.

Note: As shown in the example, each value must be entered on a separate line.

NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example (**4, 5**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line.

- **is not null** Finds all non-blank values. Click here for an example.

  Example: `ciaValue is not null` matches any incident with a varbind that contains a value.

- **is null** Finds all blank values. Click here for an example.

  Example: `ciaValue is null` matches any incident with a varbind that does not contain a value.

- **like** Finds matches using the syntax defined for java regular expressions. See the Pattern (Java Platform SE6) API documentation at : `http://download.oracle.com/javase/6/docs/api/java/util/regex/Pattern.html` for more information. Click here for more information.

  The period asterisk (.*) characters mean *any number of characters of any type at this location*.

  The period (.) character means *any single character of any type at this location*.

  Note: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below.

  Example:

  `ciaName like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that begins with 1.3.6.1.4.1.9.9 and

Interface Settings Dampening Configuration Attributes , continued

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td></td><td>

Payload Filter Editor Components, continued

<table>
<tr><th>Attribute</th><th>Description</th></tr>
<tr><td></td><td>

(optionally) ends with any number of characters.

`ciaValue like .*Chicago.*` finds all traps or events that contain a varbind value that includes the string **Chicago**.

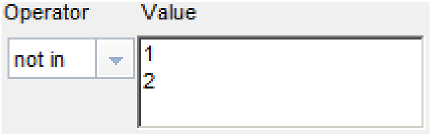
- **not between** Finds all values except those between the two values specified. Click here for an example.

  Example: `ciaValue not between 5 8` matches an incident that contains a varbind with the values less than **5** or greater than **8** .

- **not in** Finds all values except those included in the list of values. Click here for an example.

  Example:

  `ciaValue not in`

  Operator: not in
  Value:
  1
  2

  matches any incident that contains a varbind with values other than **1** and **2**.

  Note: As shown in the example, each value must be entered on a separate line.

  NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example, (**1, 2**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line.

- **not like** Finds all that do not have the values specified using the syntax defined for Java regular expressions. See the Pattern (Java Platform SE6) API documentation at: `http://download.oracle.com/javase/6/docs/api/java/util/regex/Pattern.html` for more information. Click here for an example.

  The period asterisk (.*) characters mean *any number of characters of any type at this location*.

  The period (.) character means *any single character of any type at this location*.

  Note: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed

</td></tr>
</table>

</td></tr>
</table>

Interface Settings Dampening Configuration Attributes , continued

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td></td><td>

<b>Payload Filter Editor Components, continued</b>

<table>
<tr><th>Attribute</th><th>Description</th></tr>
<tr><td></td><td>below.

Example:

<code>ciaName not like \Q.1.3.6.1.4.1.9.9\E.*</code> matches any incident that contains a varbind name value that does not begin with <b>.1.3.6.1.4.1.9.9</b> and (optionally) ends with any number of characters.

<code>ciaValue not like .*Chicago.*</code> finds all traps or events that do not contain a varbind value that includes the string <b>Chicago</b>.</td></tr>
<tr><td>Value</td><td>The value for which you want NNMi to search.

Note the following:<ul><li>The values you enter are case sensitive.</li><li>NNMi displays a variable number of value fields depending on the Operator selected. For example, the <code>between</code> Operator causes two value fields to be displayed.</li><li>The <code>between, in</code> and <code>not in</code> operators require that each value be entered on a separate line.</li></ul></td></tr>
</table>

<b>Payload Filter Editor Buttons</b>

<table>
<tr><th>Button</th><th>Description</th></tr>
<tr><td>Append</td><td>Appends the current expression (Attribute, Operator,and Value) to the selected expression already included in the filter string.</td></tr>
<tr><td>Insert</td><td>Inserts the current expression (Attribute, Operator,and Value) in front of the cursor location within the Filter String.</td></tr>
<tr><td>Replace</td><td>Replaces the selected expression with the expression displayed in the Attribute, Operator, and Value fields.</td></tr>
<tr><td>AND</td><td>Inserts the AND Boolean Operator in the selected cursor location.

<b>Note</b>: View the expression displayed under <b>Filter String</b> to see the logic of the expression as it is created.</td></tr>
<tr><td>OR</td><td>Inserts the OR Boolean Operator in the current cursor location.

<b>Note</b>: View the expression displayed under <b>Filter String</b> to see the logic of the expression as it is created.</td></tr>
<tr><td>NOT</td><td>Can be used in any part of the Filter String to specify that NNMi</td></tr>
</table>

</td></tr>
</table>

Interface Settings Dampening Configuration Attributes , continued

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td></td><td>
<b>Payload Filter Editor Buttons, continued</b>
<table>
<tr><th>Button</th><th>Description</th></tr>
<tr><td></td><td>should exclude interfaces with values that pass the expression that immediately follows the <code>NOT</code>.
For example, when evaluating the following expression, NNMi includes interfaces with (interface description) <code>ifDesc</code> containing <b>VLAN</b>, and excludes any Interfaces that have <b>VLAN10</b> for the (interface name) <code>ifName</code> value:
<code>(ifDesc like VLAN AND NOT (ifName=VLAN10))</code>
<b>Note</b>: View the expression displayed under Filter String to see the logic of the expression as it is created .</td></tr>
<tr><td>EXISTS</td><td>Used for filters that include Capabilities or Custom Attribute names and values in the Filer String.
Indicates that you want NNMi to consider interfaces that do not have any Capabilities or Custom Attributes when evaluating the Filter String.
<b>Note</b>: If you include Capabilities or Custom Attribute names and values in the Filer String, but do not use <b>EXISTS</b> or <b>NOT EXISTS</b>, NNMi excludes from its search interfaces that do not include Capabilities or Custom Attributes.
For example, when evaluating the following Filter String, NNMi includes interfaces with (interface description) <code>ifDesc</code> containing <b>VLAN</b>, as well as any Interfaces Custom Attribute Role value is <b>LAN Connection to Oracle Server</b>:
<code>(ifDesc like VLAN OR EXISTS((customAttrName=Role AND customAttrValue=LAN Connection to Oracle Server)))</code>
<b>Note</b>: View the expression displayed under Filter String to see the logic of the expression as it is created.</td></tr>
<tr><td>NOT EXISTS</td><td>Used for filters that include Capabilities or Custom Attribute names and values in the Filer String. Indicates that you want NNMi to consider interfaces that do not have any Capabilities or Custom Attributes when evaluating the Filter String, but exclude the interfaces that match the expression that follows the <b>NOT EXISTS</b>.
<b>Note</b>: If you include Capabilities or Custom Attribute names and values in the Filer String, but do not use <b>EXISTS</b> or <b>NOT EXISTS</b>, NNMi excludes from its search interfaces that do not include Capabilities or Custom Attributes.</td></tr>
</table>
</td></tr>
</table>

Interface Settings Dampening Configuration Attributes , continued

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td></td><td>
<b>Payload Filter Editor Buttons, continued</b>
<table>
<tr><th>Button</th><th>Description</th></tr>
<tr><td></td><td>For example, when evaluating the following expression, NNMi includes interfaces with (interface description) ifDesc containing <b>VLAN</b>, and excludes any Interfaces that have the Custom Attribute Role and that Role value is <b>LAN Connection to Oracle Server</b>:
(ifDesc like VLAN OR NOT EXISTS ((customAttrName=Role AND customAttrValue=LAN Connection to Oracle Server)))
<b>Note</b>: View the expression displayed under Filter String to see the logic of the expression as it is created.</td></tr>
<tr><td>Delete</td><td>Deletes the selected expression.
<b>Note</b>: If the Boolean Operator is selected, the Payload Filter Editor deletes all expressions associated with the Boolean Operator.</td></tr>
</table>
</td></tr>
</table>

Configure Incident Actions for an Interface Group (Management Events)

Note: Interface Settings override any other Actions settings for this incident, including those from the Node Settings tab.

For information about each Interface Settings tab:

NNMi enables you to configure incident actions based on a Source Object's participation in an Interface Group.

You can configure actions to automatically run at any point in the incident lifecycle. For example, you might want to configure an action to occur when an incident of the type you are configuring is generated (**Registered**). When an incident is generated, you might want to automatically open a trouble ticket or send email or page your network operator. After the incident is **Closed**, you might want to automatically close the trouble ticket.

Note: Your actions will not be executed until you enable the Actions configuration by either clicking Enable ☑ on the Actions tab or using the **Actions** → **Enable Configuration** option.

You can configure actions for incidents generated from SNMP Trap Incidents, Syslog Messages Incidents and the NNMi Management Events Incidents. Any time an incident configuration changes, the action directory is rescanned and any Jython files are reloaded to the NNMi database. See "Lifecycle Transition Action Form (Management Events)" on page 1207 for more information about the actions directory.

Tip: Copy any required Jython files to the NNMi actions directory before you configure an incident action. New or updated actions are loaded into NNMi only when an incident configuration is updated or created. See "Lifecycle Transition Action Form (Management Events)" on page 1207 for the location of the NNMi action directory.

When the action runs, output is logged to the `incidentActions.*.*.log` file. See "Verify that NNMi Services are Running" on page 85 for more information about log files and where they are located.

To configure an automatic action for an incident:

1. Navigate to the **Management Event Configuration** form.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Expand the **Incidents** folder.
   c. Select **Management Event Configurations**.
   d. Do one of the following:
      i. To create a new incident configuration, click the ✳ New icon.
      ii. To edit an existing incident configuration, select a row, click the 📂 Open icon, and continue.
2. Select the **Interface Settings** tab.
3. Do one of the following:
   a. To create a new configuration, click the ✳ New icon.
   b. To edit an existing configuration, select a row, click the 📂 Open icon, and continue.
4. Make sure you configure the basic Interface Setting behavior. See "Configure Interface Settings for a Management Event Incident" on page 1098 for more information.
5. Select the **Actions** tab.
6. From the **Lifecycle Actions** table toolbar, do one of the following:
   - To create an Action configuration, click the ✳ New icon, and continue.
   - To edit an Action configuration, select a row, click the 📂 Open icon, and continue.
   - To delete an Action configuration, select a row, and click the ✖ Delete icon.
7. In the "Lifecycle Transition Action Form (Management Events)" on page 1207, provide the required information.
8. Click 💾 **Save and Close** to save your changes and return to the previous form.

   The next time the lifecycle changes, NNMi launches the action associated with the lifecycle for the incident of that type.

Configure a Payload Filter for an Incident Action (Interface Settings) (Management Events)

The Payload Filter Editor enables you to create expressions that further refine the filters used to select the incidents that cause the configured action to run. Make sure to design any complex Payload Filters offline as a Boolean expression first. This method can help to minimize errors when entering your expressions using the Payload Filter editor.

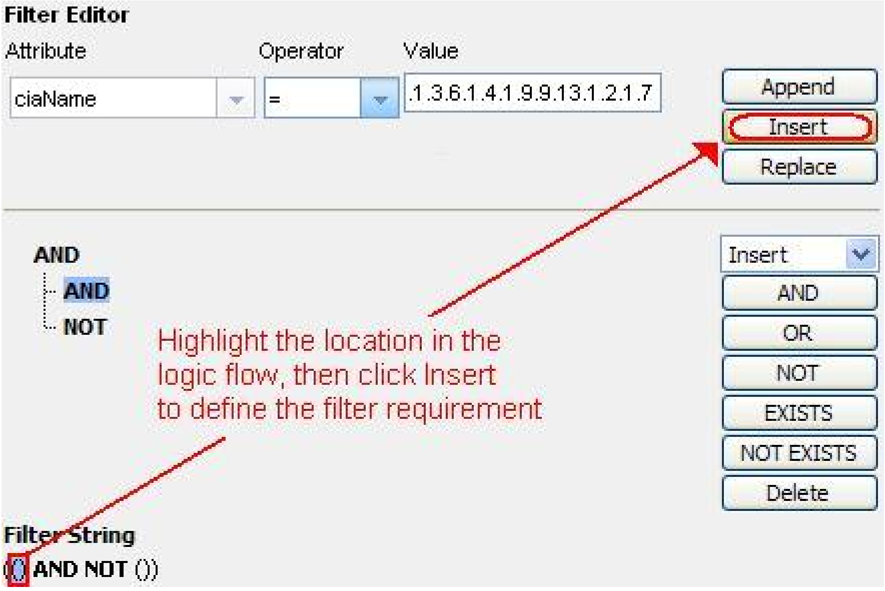
To create a Payload Filter expression:

1. Navigate to the **Management Event Configuration** form:
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Expand the **Incidents** folder.
   c. Select **Management Event Configurations** .
   d. Do one of the following:
      i. To create an incident configuration, click the ✱ New icon, and continue.
      ii. To edit an incident configuration, select a row, click the 🗁 Open icon, and continue.
      iii. To delete an incident configuration, select a row, and click the ✖ Delete icon.
2. Select the **Interface Settings** tab.
3. Do one of the following:
   a. To create a new configuration, click the ✱ New icon.
   b. To edit an existing configuration, double-click the row representing the configuration you want to edit.
4. Make sure you configure the basic Interface Setting behavior. See "Configure Interface Settings for a Management Event Incident" on page 1098 for more information.
5. Select the **Actions** tab.
6. Do one of the following:
   a. To create an Action configuration, click the ✱ New icon, and continue.
   b. To edit an Action configuration, double-click the row representing the configuration you want to edit, and continue.
   c. To delete an Action configuration, select a row, and click the ✖ Delete icon.
7. Make sure the Action settings are configured. See "Configure Incident Actions for an Interface Group (Management Events)" on page 1127 for more information.
8. Select the **Payload Filter** tab.
9. Define your Payload Filter (see table). Also see "Guidelines for Creating a Payload Filter".
   a. Plan out the logic needed for your Filter String.
   b. Use the buttons on the bottom half of the Additional Filters Editor to establish the logic structure.

      For example, to establish the following structure, select **Insert**, then click **AND**, then **NOT**, and then **AND** a second time:

      ```
      (( ) AND NOT ( ))
      ```
   c. Now place your cursor in a location within the displayed Filter String, and use the top half of the filter editor to define the parameters of the highlighted filter requirement.

For example, select a set of parentheses and use the Insert button to specify the filter requirement within those parentheses:



Attribute: ciaName
Operator: =
Value: .1.3.6.1.4.1.9.9.13.1.2.1.7

Highlight the location in the logic flow, then click Insert to define the filter requirement

Filter String
() AND NOT ()

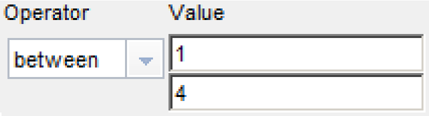
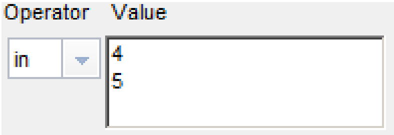
10. Click **Save and Close**.

11. Click **Save and Close** to save your changes and return to the previous form.

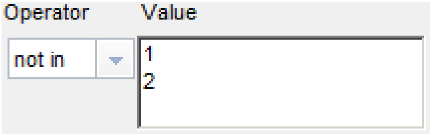
Payload Filter Editor Components

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: • ciaName • ciaValue
Operator	Valid operators are described below. • **=** Finds all values equal to the value specified. Click here for an example. Example: `ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7` matches any incident that contains a varbind with the name value **.1.3.6.1.4.1.9.9.13.1.2.1.7**. • **!=** Finds all values not equal to the value specified. Click here for an example. Example: `ciaName != .1.3.6.1.4.1.9.9.13.1.2.1.7` matches any incident that contains a varbind with the name value other than **1.3.6.1.4.1.9.9.13.1.2.1.7**. • **<** Finds all values less than the value specified. Click here for an example. Example: `ciaValue < 6` matches any incident with a varbind value less than **6**. • **<=** Finds all values less than or equal to the value specified. Click here for an example. Example: `ciaValue <= 6` matches any incident with a varbind value less than

Payload Filter Editor Components, continued

Attribute	Description
	or equal to **6**. - **>** Finds all values greater than the value specified. Click here for an example. Example: `ciaValue > 4` matches any incident with a varbind value greater than **4**. - **>=** Finds all values greater than or equal to the value specified. Click here for an example. Example: `ciaValue >= 4` matches any incident that contains a varbind with values greater than or equal to **4**. - **between** Finds all traps or events that include a varbind with a value equal to and between the two values specified. Click here for an example. Example: `ciaValue between`  Operator: between Value: 1 4 matches any incident that contains a varbind value equal to or greater than **1** and equal to or less than **4**. **Note**: As shown in the example, each value must be entered on a separate line. - **in** Finds any match to at least one value in a list of values. Click here for an example. Example: `ciaValue in`  Operator: in Value: 4 5 matches any incident with a varbind value of either **4** or **5**. **Note**: As shown in the example, each value must be entered on a separate line. NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example (**4, 5**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line. - **is not null** Finds all non-blank values. Click here for an example. Example: `ciaValue is not null` matches any incident with a varbind that contains a value. - **is null** Finds all blank values. Click here for an example. Example: `ciaValue is null` matches any incident with a varbind that does not have a value.

Payload Filter Editor Components, continued

Attribute	Description
	• **like** Finds matches using wildcard characters. Click here for more information about using wildcard characters. The period asterisk (.*) characters mean *any number of characters of any type at this location*. The period (.) character means *any single character of any type at this location*. **Note**: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below. Examples: `ciaName like  \Q.1.3.6.1.4.1.9.9\E.*` finds all traps or events that contain varbind names that begin with **.1.3.6.1.4.1.9.9** and (optionally) end with any number of characters. `ciaValue like .*Chicago.*` finds all traps or events that contain a varbind value that includes the string **Chicago**. • **not between** Finds all values except those between the two values specified. Click here for an example. Example: `ciaValue not between 5 8` matches an incident that contains a varbind with the values less than **5** or greater than **8**. • **not in** Finds all values except those included in the list of values. Click here for an example. Example: `ciaValue not in`  Operator: not in Value: 1 2 matches any incident that contains a varbind with values other than **1** and **2**. **Note**: As shown in the example, each value must be entered on a separate line. NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example, (**1, 2**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line. • **not like** Finds all that do not have the values specified (using wildcard strings). Click here for an example. The period asterisk (.*) characters mean *any number of characters of any type at this location*. The period (.) character means *any single character of any type at this location*. **Note**: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below.

Payload Filter Editor Components, continued

Attribute	Description
	Example: `ciaName not like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that does not begin with **.1.3.6.1.4.1.9.9**. `ciaValue not like .*Chicago.*` finds all traps or events that do not contain a varbind value that includes the string **Chicago**.
Value	The value for which you want NNMi to search. Note the following: • The values you enter are case sensitive. • NNMi displays a variable number of value fields depending on the Operator selected. For example, the `between` Operator causes two value fields to be displayed. • The `between, in` and `not in` operators require that each value be entered on a separate line.

Additional Filters Editor Buttons

Button	Description
Append	Appends the current expression (Attribute, Operator,and Value) to the selected expression already included in the Filter String.
Insert	Inserts the current expression (Attribute, Operator,and Value) in front of the cursor location within the Filter String.
Replace	Replaces the selected expression with the expression displayed in the Attribute, Operator, and Value fields.
AND	Appends, inserts, or replaces the AND Boolean Operator in the selected cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
OR	Appends, Inserts, or replaces the OR Boolean Operator in the current cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
NOT	Can be used in any part of the Filter String to specify that NNMi should exclude interfaces with values that pass the expression that immediately follows the `NOT`. For example, when evaluating the following expression, NNMi includes interfaces with (interface description) `ifDesc` containing **VLAN**, and excludes any Interfaces that have **VLAN10** for the (interface name) `ifName` value:

Additional Filters Editor Buttons, continued

Button	Description
	`(ifDesc like VLAN AND NOT (ifName=VLAN10))` **Note**: View the expression displayed under Filter String to see the logic of the expression as it is created .
EXISTS	Used for filters that include Capabilities or Custom Attribute names and values in the Filer String. Indicates that you want NNMi to consider interfaces that do not have any Capabilities or Custom Attributes when evaluating the Filter String. **Note**: If you include Capabilities or Custom Attribute names and values in the Filer String, but do not use **EXISTS** or **NOT EXISTS**, NNMi excludes from its search interfaces that do not include Capabilities or Custom Attributes. For example, when evaluating the following Filter String, NNMi includes interfaces with (interface description) `ifDesc` containing **VLAN**, as well as any Interfaces Custom Attribute Role value is **LAN Connection to Oracle Server**: `(ifDesc like VLAN OR EXISTS((customAttrName=Role AND customAttrValue=LAN Connection to Oracle Server)))` **Note**: View the expression displayed under Filter String to see the logic of the expression as it is created.
NOT EXISTS	Used for filters that include Capabilities or Custom Attribute names and values in the Filer String. Indicates that you want NNMi to consider interfaces that do not have any Capabilities or Custom Attributes when evaluating the Filter String, but exclude the interfaces that match the expression that follows the **NOT EXISTS**. Note: If you include Capabilities or Custom Attribute names and values in the Filer String, but do not use **EXISTS** or **NOT EXISTS**, NNMi excludes from its search interfaces that do not include Capabilities or Custom Attributes. For example, when evaluating the following expression, NNMi includes interfaces with (interface description) `ifDesc` containing **VLAN**, and excludes any Interfaces that have the Custom Attribute Role and that Role value is **LAN Connection to Oracle Server**: `(ifDesc like VLAN OR NOT EXISTS((customAttrName=Role AND customAttrValue=LAN Connection to Oracle Server)))` **Note**: View the expression displayed under Filter String to see the logic of the expression as it is created.
Delete	Deletes the selected expression. **Note**: If the Boolean Operator is selected, the Additional Filters Editor deletes all expressions associated with the Boolean Operator.

Configure Node Settings for a Management Event Incident

Note: Node Settings override any other Suppression, Enrichment, Dampen, Action, or Diagnostics Selections configuration settings, except those configured on the Interface Settings tab.

NNMi enables you to apply an incident configuration to a Source Node based on the Source Node's participation in a Node Group. If the Source Node is not a member of the Node Group specified, the incident is neither displayed nor stored in the NNMi database.

Tip: See "Create Node Groups" on page 295 for more information about Node Groups.

For information about each Node Settings tab:

For information about each Management Events tab:

To apply an incident configuration to a Source Node based on the Source Node's Node Group:

1. Navigate to the **Management Event Configuration** form:
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Expand the **Incidents** folder.
   c. Select **Management Event Configurations** .
   d. Do one of the following:
      i. To create an incident configuration, click the ✱ New icon, and continue.
      ii. To edit an incident configuration, select a row, click the Open icon, and continue.
      iii. To delete an incident configuration, select a row, and click the ✖ Delete icon.
2. Select the **Node Settings** tab.
3. Do one of the following:
   a. To create a new configuration, click the ✱ New icon.
   b. To edit an existing configuration, double-click the row representing the configuration you want to edit.
4. Configure the desired Node Settings (see table).
5. Click **Save and Close** to save your changes and return to the previous form.

Node Group Attributes

Name	Description
Node Group	Click the Lookup icon and select Quick Find to select the Node Group you want to use. See "Use the Quick Find Window" on page 41 for more information about using Quick Find.
Ordering	Determines the priority order for those nodes that appear in multiple Node Groups. The lower the number, the higher the priority. For example, **1** is the highest priority. If a

Node Group Attributes , continued

Name	Description
	node is in multiple Node Groups and more than one of those Node Groups have been specified in an incident configuration, only the incident configuration with the highest priority will be applied to the node.
Enable	Use this attribute to temporarily disable an incident's suppression settings: **Enable** □ = Temporarily disable the selected configuration. **Enable** ☑ = Enable the selected configuration.

Configure Incident Suppression Settings for a Node Group (Management Events)

Note: Node Settings override any other Suppression settings for this incident, except those configured on the Interface Settings tab.

NNMi enables you to suppress a specified incident configuration based on the Source Node's participation in a Node Group.

Note: You can also suppress the incident configuration based on the Source Object's participation in an Interface Group. See "Configure Incident Suppression Settings for an Interface Group (Management Events)" on page 1099 for more information.

Tip: See "Create Node Groups" on page 295for more information about Node Groups.

For information about each Node Settings tab:

To suppress an incident configuration based on a Node Group:

1. Navigate to the **Management Event Configuration** form:
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Expand the **Incidents** folder.
   c. Select **Management Event Configurations** .
   d. Do one of the following:
      i. To create an incident configuration, click the ✱ New icon, and continue.
      ii. To edit an incident configuration, select a row, click the Open icon, and continue.
      iii. To delete an incident configuration, select a row, and click the ✖ Delete icon.
2. Select the **Node Settings** tab.
3. Make sure you configure the basic Node Setting behavior. See "Configure Node Settings for a Management Event Incident" on the previous page for more information.
5. Select the **Suppression** tab.
6. Configure the desired Suppression behavior (see table).
7. Click **Save and Close** to save your changes and return to the previous form.

Node Settings Suppression Attributes

Name	Description
Enable	Use this attribute to temporarily disable an incident's suppression settings: **Enable** □ = Temporarily disable the selected configuration. **Enable** ☑ = Enable the selected configuration.

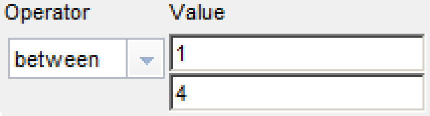
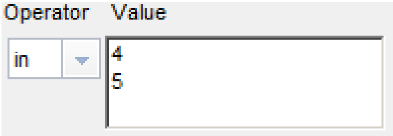
Payload Filter	The Payload Filter Editor enables you to create expressions that further refine the filters used to select the incidents to be suppressed, enriched, or dampened. Make sure to design any complex Payload Filters offline as a Boolean expression first. This method can help to minimize errors when entering your expressions using the Payload Filter editor.
	When creating a Payload Filter, note the following:
	• Payload Filter expressions for the `like` and `not like` operators use the syntax defined for java regular expressions (java.util.regex Pattern class).
	• You must use a `ciaName` that already exists in the trap or event you are configuring.
	• Each set of expressions associated with a Boolean Operator is treated as if it were enclosed in parentheses and evaluated together.
	• View the expression displayed under **Filter String** to see the logic of the expression as it is created.
	• The `AND` and `OR` Boolean Operators must contain at least two expressions as shown in the example below.
	The following example filters incidents on voltage state:
	`AND`
	`    ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7`
	`    ciaValue = 5`
	NNMi evaluates the expression above as follows:
	`(ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7 AND ciaValue = 5)`
	NNMi finds all incidents with a varbind `.1.3.6.1.4.1.9.9.13.1.2.1.7` value of **5**.
	• The placement of your cursor and the subsequent text that is selected is important when performing operations using the Payload Filter Editor. For example, you append to, replace, or change the indentation of the expression that is selected.
	• The placement of your cursor and the subsequent text that is selected is especially important when adding your Boolean operators.
	• You can include more than one varbind in the same Payload Filter expression as shown in the following example:
	`((ciaName like \Q.1.3.6.1.4.1.9.9\E.* AND ciaValue = 25) AND (ciaName like \Q.1.3.6.1.2.1.2.2.1.1.3\E.* AND ciaValue = 3))`

Node Settings Suppression Attributes , continued

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td></td><td>
<p>In this example, a given trap must meet each of the following criteria:</p>
<ul>
<li>Contain a varbind whose Object Identifier (OID) matches the regular expression <code>\Q.1.3.6.1.4.1.9.9\E.*</code> and has a value of <code>25</code>.</li>
<li>Contain a varbind whose OID matches the regular expression <code>\Q.1.3.6.1.2.1.2.2.1.1.3\E.*</code> and has a value of <code>3</code>.</li>
</ul>
<p><b>Payload Filter Editor Components</b></p>
<table>
<tr><th>Attribute</th><th>Description</th></tr>
<tr><td>Attribute</td><td>
<p>The attribute name on which NNMi searches. Filterable attributes include the following:</p>
<ul>
<li>ciaName</li>
<li>ciaValue</li>
</ul>
</td></tr>
<tr><td>Operator</td><td>
<p>Valid operators are described below.</p>
<ul>
<li><b>=</b> Finds all values equal to the value specified. Click here for an example.
<p>Example: <code>ciaName=.1.3.6.1.4.1.9.9.13.1.2.1.7</code> matches any incident that contains a varbind with the name value of <b>.1.3.6.1.4.1.9.9.13.1.2.1.7</b>.</p></li>
<li><b>!=</b> Finds all values not equal to the value specified. Click here for an example.
<p>Example: <code>ciaName! = .1.3.6.1.4.1.9.9.13.1.2.1.7</code> matches any incident that contains a varbind with a name value other than <b>1.3.6.1.4.1.9.9.13.1.2.1.7</b>.</p></li>
<li><b>&lt;</b> Finds all values less than the value specified. Click here for an example.
<p>Example: <code>ciaValue &lt; 6</code> matches any incident that contains a varbind with a value less than <b>6</b>.</p></li>
<li><b>&lt;=</b> Finds all values less than or equal to the value specified. Click here for an example.
<p>Example: <code>ciaValue &lt;= 6</code> matches any incident that contains a varbind with a value less than or equal to <b>6</b>.</p></li>
<li><b>&gt;</b> Finds all values greater than the value specified. Click here for an example.
<p>Example: <code>ciaValue &gt; 4</code> matches any incident that contains a varbind with a value greater than <b>4</b>.</p></li>
<li><b>&gt;=</b> Finds all values greater than or equal to the value specified. Click here for an example.</li>
</ul>
</td></tr>
</table>
</td></tr>
</table>

Node Settings Suppression Attributes , continued

Name	Description
	Payload Filter Editor Components, continued

Attribute	Description
	Example: `ciaValue >= 4` matches any incident that contains a varbind with values greater than or equal to **4**. - **between** Finds all values equal to and between the two values specified. Click here for an example. Example: `ciaValue between`  Operator: between Value: 1 4 matches any incident that contains a varbind value equal to or greater than **1** and equal to or less than **4**. **Note**: As shown in the example, each value must be entered on a separate line. - **in** Finds any match to at least one value in a list of values. Click here for an example. Example: `ciaValue in`  Operator: in Value: 4 5 matches any incident with a varbind value of either **4** or **5**. **Note**: As shown in the example, each value must be entered on a separate line. NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example (**4, 5**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line. - **is not null** Finds all non-blank values. Click here for an example. Example: `ciaValue is not null` matches any incident with a varbind that contains a value. - **is null** Finds all blank values. Click here for an example. Example: `ciaValue is null` matches any incident with a varbind that does not contain a value.

Node Settings Suppression Attributes , continued

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td></td><td>

Payload Filter Editor Components, continued

<table>
<tr><th>Attribute</th><th>Description</th></tr>
<tr><td></td><td>

- **like** Finds matches using the syntax defined for java regular expressions. See the Pattern (Java Platform SE6) API documentation at : `http://download.oracle.com/javase/6/docs/api/java/util/regex/Pattern.html` for more information. Click here for more information.

  The period asterisk (.*) characters mean *any number of characters of any type at this location*.

  The period (.) character means *any single character of any type at this location*.

  Note: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below.

  Example:

  `ciaName like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that begins with 1.3.6.1.4.1.9.9 and (optionally) ends with any number of characters.

  `ciaValue like .*Chicago.*` finds all traps or events that contain a varbind value that includes the string **Chicago**.

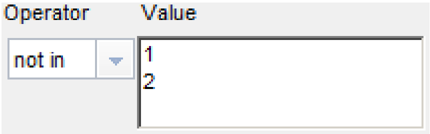
- **not between** Finds all values except those between the two values specified. Click here for an example.

  Example: `ciaValue not between 5 8` matches an incident that contains a varbind with the values less than **5** or greater than **8** .

- **not in** Finds all values except those included in the list of values. Click here for an example.

  Example:

  `ciaValue not in`

  Operator: not in
  Value: 1
2

  matches any incident that contains a varbind with values other than **1** and **2**.

  Note: As shown in the example, each value must be entered on a separate line.

</td></tr>
</table>

</td></tr>
</table>

Node Settings Suppression Attributes , continued

Name	Description

Payload Filter Editor Components, continued

Attribute	Description
	NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example, (**1, 2**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line. • **not like** Finds all that do not have the values specified using the syntax defined for Java regular expressions. See the Pattern (Java Platform SE6) API documentation at: `http://download.oracle.com/javase/6/docs/api/java/util/regex/Pattern.html` for more information. Click here for an example. The period asterisk (.*) characters mean *any number of characters of any type at this location*. The period (.) character means *any single character of any type at this location*. **Note**: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below. Example: `ciaName not like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that does not begin with **.1.3.6.1.4.1.9.9** and (optionally) ends with any number of characters. `ciaValue not like .*Chicago.*` finds all traps or events that do not contain a varbind value that includes the string **Chicago**.
Value	The value for which you want NNMi to search. Note the following: • The values you enter are case sensitive. • NNMi displays a variable number of value fields depending on the Operator selected. For example, the `between` Operator causes two value fields to be displayed. • The `between, in` and `not in` operators require that each value be entered on a separate line.

Node Settings Suppression Attributes , continued

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td></td><td>

Payload Filter Editor Buttons

<table>
<tr><th>Button</th><th>Description</th></tr>
<tr><td>Append</td><td>Appends the current expression (Attribute, Operator,and Value) to the selected expression already included in the filter string.</td></tr>
<tr><td>Insert</td><td>Inserts the current expression (Attribute, Operator,and Value) in front of the cursor location within the Filter String.</td></tr>
<tr><td>Replace</td><td>Replaces the selected expression with the expression displayed in the Attribute, Operator, and Value fields.</td></tr>
<tr><td>AND</td><td>Inserts the AND Boolean Operator in the selected cursor location.
<b>Note</b>: View the expression displayed under <b>Filter String</b> to see the logic of the expression as it is created.</td></tr>
<tr><td>OR</td><td>Inserts the OR Boolean Operator in the current cursor location.
<b>Note</b>: View the expression displayed under <b>Filter String</b> to see the logic of the expression as it is created.</td></tr>
<tr><td>NOT</td><td>Can be used in any part of the Filter String to specify that NNMi should exclude interfaces with values that pass the expression that immediately follows the <code>NOT</code>.
For example, when evaluating the following expression, NNMi includes interfaces with (interface description) <code>ifDesc</code> containing <b>VLAN</b>, and excludes any Interfaces that have <b>VLAN10</b> for the (interface name) <code>ifName</code> value:
<code>(ifDesc like VLAN AND NOT (ifName=VLAN10))</code>
<b>Note</b>: View the expression displayed under Filter String to see the logic of the expression as it is created .</td></tr>
<tr><td>EXISTS</td><td>Used for filters that include Capabilities or Custom Attribute names and values in the Filer String.
Indicates that you want NNMi to consider interfaces that do not have any Capabilities or Custom Attributes when evaluating the Filter String.
<b>Note</b>: If you include Capabilities or Custom Attribute names and values in the Filer String, but do not use <b>EXISTS</b> or <b>NOT EXISTS</b>, NNMi excludes from its search interfaces that do not include Capabilities or Custom Attributes.
For example, when evaluating the following Filter String, NNMi includes interfaces with (interface description) <code>ifDesc</code> containing</td></tr>
</table>

</td></tr>
</table>

Node Settings Suppression Attributes , continued

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td></td><td>
<b>Payload Filter Editor Buttons, continued</b>
<table>
<tr><th>Button</th><th>Description</th></tr>
<tr><td></td><td><b>VLAN</b>, as well as any Interfaces Custom Attribute Role value is <b>LAN Connection to Oracle Server</b>:
<code>(ifDesc like VLAN OR EXISTS((customAttrName=Role AND customAttrValue=LAN Connection to Oracle Server)))</code>
<b>Note</b>: View the expression displayed under Filter String to see the logic of the expression as it is created.</td></tr>
<tr><td>NOT EXISTS</td><td>Used for filters that include Capabilities or Custom Attribute names and values in the Filer String. Indicates that you want NNMi to consider interfaces that do not have any Capabilities or Custom Attributes when evaluating the Filter String, but exclude the interfaces that match the expression that follows the <b>NOT EXISTS</b>.
<b>Note</b>: If you include Capabilities or Custom Attribute names and values in the Filer String, but do not use <b>EXISTS</b> or <b>NOT EXISTS</b>, NNMi excludes from its search interfaces that do not include Capabilities or Custom Attributes.
For example, when evaluating the following expression, NNMi includes interfaces with (interface description) <code>ifDesc</code> containing <b>VLAN</b>, and excludes any Interfaces that have the Custom Attribute Role and that Role value is <b>LAN Connection to Oracle Server</b>:
<code>(ifDesc like VLAN OR NOT EXISTS ((customAttrName=Role AND customAttrValue=LAN Connection to Oracle Server)))</code>
<b>Note</b>: View the expression displayed under Filter String to see the logic of the expression as it is created.</td></tr>
<tr><td>Delete</td><td>Deletes the selected expression.
<b>Note</b>: If the Boolean Operator is selected, the Payload Filter Editor deletes all expressions associated with the Boolean Operator.</td></tr>
</table>
</td></tr>
</table>

Configure Incident Enrichment Settings for a Node Group (Management Events)

Note: Node Settings override any other Enrichment settings for this incident, except those configured on the Interface Settings tab.

NNMi enables you to enhanced a specified incident configuration based on the Source Node's participation in a Node Group. The types of items you can fine tune and enhance for a selected incident configuration, include:

- Category
- Family
- Severity
- Priority
- Correlation Nature
- Message
- Assigned To

Note: You can also enhance the incident configuration based on the Source Object's participation in an Interface Group. See "Configure Incident Enrichment Settings for an Interface Group (Management Events)" on page 1107 for more information.

Tip: See "Create Node Groups" on page 295 for more information about Node Groups.

For information about each Node Settings tab:

For information about each Enrichment tab:

To configure Enrichment settings for a Node Group:

1. Navigate to the **Management Event Configuration** form:
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Select the **Incidents** folder.
   c. Select **Management Event Configurations** .
   d. Do one of the following:
      i. To create an incident configuration, click the New icon, and continue.
      ii. To edit an incident configuration, select a row, click the Open icon, and continue.
      iii. To delete an incident configuration, select a row, and click the Delete icon.
2. Select the **Node Settings** tab.
3. Do one of the following:
   a. To create a new configuration, click the New icon.
   b. To edit an existing configuration, select a row, click the Open icon, and continue.
4. Make sure you configure the basic Node Setting behavior. See "Configure Node Settings for a Management Event Incident" on page 1135 for more information.
5. Select the **Enrichment** tab.
6. Do one of the following:
   a. To create an Enrichment configuration, click the New icon and continue.
   b. To edit an Enrichment configuration, select a row, click the Open icon, and continue.

c. To delete an Enrichment configuration, select a row and click the ✖ Delete icon.

7. Configure the desired Enrichment behavior (see the "Enrich Configuration Attributes" table)

8. Click **Save and Close** to save your changes and return to the previous form.

Node Settings Enrichment Configuration Attributes

Name	Description
Category	Use the Category attribute to customize the category for this incident configuration. Possible values include: - Accounting - Application Status - Configuration - Fault - Performance - Security - Status See "Specify Category and Family Attribute Values for Organizing Your Incidents (SNMP Trap Incident)" on page 790 for more information.
Family	Use the Family attribute to customize the Family for this incident configuration. Select from the drop-down list or create a new value. For example, some of the values provided by NNMi include: - Address - Aggregated Port (Interfaces using **Link Aggregation**[1] protocol. See Interface Form: Link Aggregation tab.) - Card - Connection - Correlation - Interface - Node
Severity	The incident Severity represents the seriousness calculated for the incident. Use the Severity attribute to specify the severity that should be assigned to the incident

[1] Protocols used on Switches to configure multiple Interfaces (Aggregation Member Interfaces) to function as if they were one (an Aggregator Interface). When two Aggregator Interfaces establish a connection, that connection is an Aggregator Layer 2 Connection. The Aggregator Layer 2 Connection appears on Layer 2 Neighbor View maps as a thick line with an Interface icon at each end (representing the Aggregator Interface). When you double-click the thick line, it converts into multiple thin lines representing the participating Aggregation Member Layer 2 Connections with their Aggregation Member Interfaces at each end of the lines.

Node Settings Enrichment Configuration Attributes , continued

Name	Description
	you are configuring. Possible values are described below: **Normal** - Indicates there are no known problems related to the associated object. This Severity is meant to be informational. Generally, no action is needed for these incidents. **Warning** - Indicates there might be a problem related to the associated object. **Minor** - Indicates NNMi has detected problems related to the associated object that require further investigation. **Major** - Indicates NNMi has detected problems related to the associated object to be resolved before they become critical. **Critical** - Indicates NNMi has detected problems related to the associated object that require immediate attention.
Priority	Used to communicate the urgency of resolving the selected incident. You control this value. NNMi sets this value to null by default. The lower the number the higher the priority. Possible values are: 5 **None** 4 **Low** 3 **Medium** 2 **High** 1 **Top** **Note**: The icons are displayed only in table views.
Correlation Nature	Use the Correlation Nature to customize the Correlation Nature for this incident configuration. Possible values include: • Info • None • Root Cause • Secondary Root Cause • Symptom • Stream Correlation • Service Impact • Dedup Stream Correlation • Rate Stream Correlation

Node Settings Enrichment Configuration Attributes , continued

Name	Description
	See Incident Form: General Tab for more information.
Message Format	When configuring an incident, specify how the incident message appears in the incident view. The string you specify in the Message Format attribute is visible in an incident view. **Note**: The incident message limit is 1024 characters. If you exceed this limit, NNMi truncates the value starting from the right. You can use any combination of default and custom attributes: "Valid Parameters for Configuring Incident Messages (Management Events)" on page 1090 "Include Custom Incident Attributes in Your Message Format (Management Events)" on page 1096
Assigned To	Use to specify the owner of any incident generated for this incident configuration. Click the Lookup icon and select Quick Find to select a valid user name. **Note**: You can also begin to type a valid user name and use the auto-complete feature to select the user name of interest.
Description	Use the Description attribute to provide additional information that you want to note about the current enhancement configuration. This description applies only to the enhancement configuration and does not appear when NNMi displays any associated incident. Type a maximum of 1024 characters. Alpha-numeric and special characters (~ ! @ # $ % ^ & * ( ) _+ -) are permitted.

Configure Custom Incident Attributes to Enrich an Incident Configuration (Node Settings) (Management Events)

The Custom Incident Attributes (CIAs) tab enables you to specify additional CIAs to be saved with an incoming incident. The persisted data might then be used as an argument to an action for the incident.

When creating a CIA for an incident configuration, you can specify any of the following values:

- Custom Attribute on the source node
- Custom Attribute on the interface (source object)

You also specify the Custom Incident Attribute name that will store this information.

For information about each Enrichment tab:

To create a Custom Incident Attribute to enrich an incident configuration:

1. Navigate to the **Management Event Configuration** form:
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Expand the **Incidents** folder.
   c. Select **Management Event Configurations**.
   d. Do one of the following:
      i. To create an incident configuration, click the New icon, and continue.
      ii. To edit an incident configuration, select a row, click the Open icon, and continue.
      iii. To delete an incident configuration, select a row, and click the Delete icon.
2. Select the **Node Settings** tab.
3. Do one of the following:
   a. To create a new configuration, click the New icon.
   b. To edit an existing configuration, select a row, click the Open icon, and continue.
4. Make sure you configure the basic Node Setting behavior. See "Configure Node Settings for a Management Event Incident" on page 1135 for more information.
5. Select the **Enrichment** tab.
6. Do one of the following:
   a. To create an Enrichment configuration, click the New icon, and continue.
   b. To edit an Enrichment configuration,select a row, click the Open icon, and continue.
   c. To delete an Enrichment configuration, select a row and click the Delete icon.
7. Make sure the Enrichment settings are configure. See "Configure Incident Enrichment Settings for a Node Group (Management Events)" on page 1143 for more information.
8. Navigate to the **Custom Incident Attributes** tab.
9. Do one of the following:
   a. To create a Custom Incident Attribute, click the New icon, and continue.
   b. To edit a Custom Incident Attribute, select a row, click the Open icon, and continue.
   c. To delete a Custom Incident Attribute, select a row and click the Delete icon.
10. Configure the Custom Incident Attribute (see the "Custom Incident Attribute" table).
11. Click **Save and Close** to save your changes and return to the previous form.

Custom Incident Attribute

Name	Description
Custom Incident Attribute Name	Name used to identify the Custom Incident Attribute you are configuring. The name limit is 255 characters. Alpha-numeric characters are permitted. No spaces or special characters (~ ! @ # $ % ^ & * ( ) _+ -) are permitted. **Note**: Make sure to note this name if you plan to filter on the value using the **Payload**

Custom Incident Attribute , continued

Name	Description
	Filter tab. See "Configure a Payload Filter to Enrich an Incident Configuration (Interface Settings) (Management Events)" on page 1113 for more information.
Type	Specifies whether you are using a Custom Attribute on a node or a Custom Attribute on an interface. Possible values are: • Node Custom Attribute • Interface Custom Attribute
Custom Attribute Name	Used to determine the value to be assigned to the Custom Incident Attribute you are configuring. Enter either of the following: • Name of the Custom Attribute on the source node • Name of the Custom Attribute on the interface (source object)

Configure a Payload Filter to Enrich an Incident Configuration (Node Settings) (Management Events)

The Payload Filter Editor enables you to create expressions that further refine the filters used to select the incidents to be enriched. Make sure to design any complex Payload Filters offline as a Boolean expression first. This method can help to minimize errors when entering your expressions using the Payload Filter editor.

To create a Payload Filter expression:

1. Navigate to the **Management Events Configuration** form:
   1. From the workspace navigation panel, select the **Configuration** workspace.
   2. Expand the **Incidents** folder.
   3. Select **Management Event Configurations**.
   4. Do one of the following:
      1. To create an incident configuration, click the New icon, and continue.
      2. To edit an incident configuration, select a row, click the Open icon, and continue.
      3. To delete an incident configuration, select a row, and click the Delete icon.
2. Select the **Node Settings** tab.
3. Do one of the following:
   1. To create a new configuration, click the New icon.
   2. To edit an existing configuration, double-click the row representing the configuration you want to edit.
4. Make sure you configure the basic Node Setting behavior. See "Configure Node Settings for a Management Event Incident" on page 1135 for more information.

5. Select the **Enrichment** tab.

6. Do one of the following:

   a. To create an Enrichment configuration, click the ✱ New icon, and continue.

   b. To edit an Enrichment configuration, select a row, click the Open icon, and continue.

   c. To delete an Enrichment configuration, select a row and click the ✖ Delete icon.

7. Make sure you configure the Enrichment settings. See "Configure Incident Enrichment Settings for a Node Group (Management Events)" on page 1143 for more information.

8. Select the **Payload Filter** tab.

9. Define your Payload Filter (see table). Also see "Guidelines for Creating a Payload Filter".

   a. Plan out the logic needed for your Filter String.

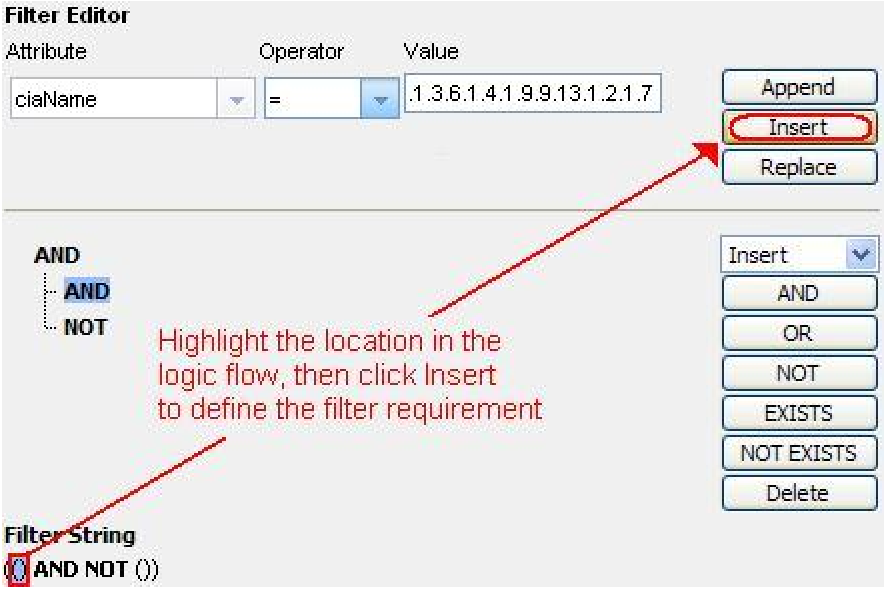
   b. Use the buttons on the bottom half of the Additional Filters Editor to establish the logic structure.

      For example, to establish the following structure, select **Insert**, then click **AND**, then **NOT**, and then **AND** a second time:

      ```
      (( ) AND NOT ( ))
      ```

   c. Now place your cursor in a location within the displayed Filter String, and use the top half of the filter editor to define the parameters of the highlighted filter requirement.

      For example, select a set of parentheses and use the Insert button to specify the filter requirement within those parentheses:



Filter Editor

Attribute: ciaName
Operator: =
Value: .1.3.6.1.4.1.9.9.13.1.2.1.7

AND
- AND
- NOT

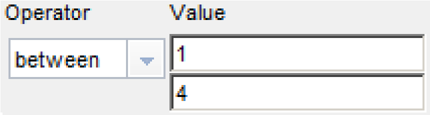
Highlight the location in the logic flow, then click Insert to define the filter requirement

Filter String
() AND NOT ()

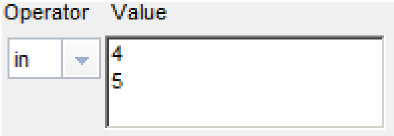
10. Click **Save and Close**.

11. Click **Save and Close** to save your changes and return to the previous form.

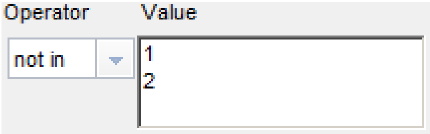
Payload Filter Editor Components

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: • ciaName • ciaValue
Operator	Valid operators are described below. • **=** Finds all values equal to the value specified. Click here for an example. Example: `ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7` matches any incident that contains a varbind with the name value **.1.3.6.1.4.1.9.9.13.1.2.1.7**. • **!=** Finds all values not equal to the value specified. Click here for an example. Example: `ciaName != .1.3.6.1.4.1.9.9.13.1.2.1.7` matches any incident that contains a varbind with the name value other than **1.3.6.1.4.1.9.9.13.1.2.1.7**. • **<** Finds all values less than the value specified. Click here for an example. Example: `ciaValue < 6` matches any incident with a varbind value less than **6**. • **<=** Finds all values less than or equal to the value specified. Click here for an example. Example: `ciaValue <= 6` matches any incident with a varbind value less than or equal to **6**. • **>** Finds all values greater than the value specified. Click here for an example. Example: `ciaValue > 4` matches any incident with a varbind value greater than **4**. • **>=** Finds all values greater than or equal to the value specified. Click here for an example. Example: `ciaValue >= 4` matches any incident that contains a varbind with values greater than or equal to **4**. • **between** Finds all traps or events that include a varbind with a value equal to and between the two values specified. Click here for an example. Example: `ciaValue between`  Operator Value between 1 4 matches any incident that contains a varbind value equal to or greater than **1** and equal to or less than **4**. **Note**: As shown in the example, each value must be entered on a separate line. • **in** Finds any match to at least one value in a list of values. Click here for an example.

Payload Filter Editor Components, continued

Attribute	Description
	Example: `ciaValue in`  Operator: in Value: 4 5 matches any incident with a varbind value of either **4** or **5**. **Note**: As shown in the example, each value must be entered on a separate line. NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example (**4, 5**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line. • **is not null** Finds all non-blank values. Click here for an example. Example: `ciaValue is not null` matches any incident with a varbind that contains a value. • **is null** Finds all blank values. Click here for an example. Example: `ciaValue is null` matches any incident with a varbind that does not have a value. • **like** Finds matches using wildcard characters. Click here for more information about using wildcard characters. The period asterisk (.*) characters mean *any number of characters of any type at this location*. The period (.) character means *any single character of any type at this location*. **Note**: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below. Examples: `ciaName like  \Q.1.3.6.1.4.1.9.9\E.*` finds all traps or events that contain varbind names that begin with **.1.3.6.1.4.1.9.9** and (optionally) end with any number of characters. `ciaValue like .*Chicago.*` finds all traps or events that contain a varbind value that includes the string **Chicago**. • **not between** Finds all values except those between the two values specified. Click here for an example. Example: `ciaValue not between 5 8` matches an incident that contains a varbind with the values less than **5** or greater than **8**. • **not in** Finds all values except those included in the list of values. Click here for an example. Example:

Payload Filter Editor Components, continued

Attribute	Description
	`ciaValue not in`  Operator not in Value 1 2 matches any incident that contains a varbind with values other than **1** and **2**. **Note**: As shown in the example, each value must be entered on a separate line. NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example, (**1, 2**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line. - **not like** Finds all that do not have the values specified (using wildcard strings). Click here for an example. The period asterisk (.*) characters mean *any number of characters of any type at this location*. The period (.) character means *any single character of any type at this location*. **Note**: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below. Example: `ciaName not like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that does not begin with **.1.3.6.1.4.1.9.9**. `ciaValue not like .*Chicago.*` finds all traps or events that do not contain a varbind value that includes the string **Chicago**.
Value	The value for which you want NNMi to search. Note the following: - The values you enter are case sensitive. - NNMi displays a variable number of value fields depending on the Operator selected. For example, the `between` Operator causes two value fields to be displayed. - The `between, in` and `not in` operators require that each value be entered on a separate line.

Additional Filters Editor Buttons

Button	Description
Append	Appends the current expression (Attribute, Operator,and Value) to the selected expression already included in the Filter String.
Insert	Inserts the current expression (Attribute, Operator,and Value) in front of the

Additional Filters Editor Buttons, continued

Button	Description
	cursor location within the Filter String.
Replace	Replaces the selected expression with the expression displayed in the Attribute, Operator, and Value fields.
AND	Appends, inserts, or replaces the AND Boolean Operator in the selected cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
OR	Appends, Inserts, or replaces the OR Boolean Operator in the current cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
NOT	Can be used in any part of the Filter String to specify that NNMi should exclude interfaces with values that pass the expression that immediately follows the `NOT`. For example, when evaluating the following expression, NNMi includes interfaces with (interface description) `ifDesc` containing **VLAN**, and excludes any Interfaces that have **VLAN10** for the (interface name) `ifName` value: `(ifDesc like VLAN AND NOT (ifName=VLAN10))` **Note**: View the expression displayed under Filter String to see the logic of the expression as it is created .
EXISTS	Used for filters that include Capabilities or Custom Attribute names and values in the Filer String. Indicates that you want NNMi to consider interfaces that do not have any Capabilities or Custom Attributes when evaluating the Filter String. **Note**: If you include Capabilities or Custom Attribute names and values in the Filer String, but do not use **EXISTS** or **NOT EXISTS**, NNMi excludes from its search interfaces that do not include Capabilities or Custom Attributes. For example, when evaluating the following Filter String, NNMi includes interfaces with (interface description) `ifDesc` containing **VLAN**, as well as any Interfaces Custom Attribute Role value is **LAN Connection to Oracle Server**: `(ifDesc like VLAN OR EXISTS((customAttrName=Role AND customAttrValue=LAN Connection to Oracle Server)))` **Note**: View the expression displayed under Filter String to see the logic of the expression as it is created.
NOT EXISTS	Used for filters that include Capabilities or Custom Attribute names and values in the Filer String. Indicates that you want NNMi to consider interfaces that do not have any Capabilities or Custom Attributes when evaluating the Filter String, but

Additional Filters Editor Buttons, continued

Button	Description
	exclude the interfaces that match the expression that follows the **NOT EXISTS**. Note: If you include Capabilities or Custom Attribute names and values in the Filer String, but do not use **EXISTS** or **NOT EXISTS**, NNMi excludes from its search interfaces that do not include Capabilities or Custom Attributes. For example, when evaluating the following expression, NNMi includes interfaces with (interface description) `ifDesc` containing **VLAN**, and excludes any Interfaces that have the Custom Attribute Role and that Role value is **LAN Connection to Oracle Server**: `(ifDesc like VLAN OR NOT EXISTS((customAttrName=Role AND customAttrValue=LAN Connection to Oracle Server)))` **Note**: View the expression displayed under Filter String to see the logic of the expression as it is created.
Delete	Deletes the selected expression. **Note**: If the Boolean Operator is selected, the Additional Filters Editor deletes all expressions associated with the Boolean Operator.

Configure Incident Dampening Settings for a Node Group (Management Events)

Note: Node Settings override any other Dampening settings for this incident, except those configured on the Interface Settings tab.

NNMi enables you to delay the following for an incident configuration based on the Source Node's participation in a Node Group:

- Execution of Incident Actions
- Execution of Diagnostics (*HP Network Node Manager iSPI Network Engineering Toolset Software \ NNM iSPI NET*)
- Appearance within Incident views in the NNMi Console

Note: You can configure the Dampening settings based on the Source Object's participation in an Interface Group. See "Configure Incident Dampening Settings for an Interface Group (Management Events)" on page 1119 for more information.

Tip: See "Create Node Groups" on page 295 for more information about Node Groups.

For information about each Node Settings tab:

When using the Dampening configuration, note the following:

- NNMi initially assigns incidents with Dampening settings configured a Lifecycle State of DAMPENED.
- After the dampen interval has passed, NNMi changes the Lifecycle State to REGISTERED.

See About the Incident Lifecycle for more information about Lifecycle State.

To configure the Dampening settings based on a Node Group:

1. Navigate to the **Management Events Configuration** form:
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Expand the **Incidents** folder.
   c. Select **Management Event Configurations** .
   d. Do one of the following:
      i. To create an incident configuration, click the ✳ New icon, and continue.
      ii. To edit an incident configuration, select a row, click the Open icon, and continue.
      iii. To delete an incident configuration, select a row, and click the ✖ Delete icon.
2. Select the **Node Settings** tab.
3. Do one of the following:
   a. To create a new configuration, click the ✳ New icon.
   b. To edit an existing configuration, select a row, click the Open icon, and continue.
4. Make sure you configure the basic Node Setting behavior. See "Configure Node Settings for a Management Event Incident" on page 1135 for more information.
5. Select the **Dampen** tab.
6. Configure the desired Dampen behavior (see table).
7. Click **Save and Close** to save your changes and return to the previous form.

Node Settings Dampen Attributes

Name	Description
Enable	Use this attribute to temporarily disable an incident's Dampening settings: **Enable** □ = Temporarily disable the selected configuration. **Enable** ☑ = Enable the selected configuration.
Hour	Specifies the number of hours to be used for the dampen interval.
Minutes	Specifies the number of minutes to be used for the dampen interval. **Note:** The maximum dampen interval is 60 minutes.
Seconds	Specifies the number of seconds to be used for the dampen interval.
Payload Filter	The Payload Filter Editor enables you to create expressions that further refine the filters used to select the incidents to be suppressed, enriched, or dampened. Make sure to design any complex Payload Filters offline as a Boolean expression first. This method can help to minimize errors when entering your expressions using the Payload Filter

Node Settings Dampen Attributes , continued

Name	Description

	editor.
	When creating a Payload Filter, note the following:
	- Payload Filter expressions for the `like` and `not like` operators use the syntax defined for java regular expressions (java.util.regex Pattern class).
	- You must use a `ciaName` that already exists in the trap or event you are configuring.
	- Each set of expressions associated with a Boolean Operator is treated as if it were enclosed in parentheses and evaluated together.
	- View the expression displayed under **Filter String** to see the logic of the expression as it is created.
	- The `AND` and `OR` Boolean Operators must contain at least two expressions as shown in the example below.
	The following example filters incidents on voltage state:
	`AND`
	    `ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7`
	    `ciaValue = 5`
	NNMi evaluates the expression above as follows:
	`(ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7 AND ciaValue = 5)`
	NNMi finds all incidents with a varbind `.1.3.6.1.4.1.9.9.13.1.2.1.7` value of **5**.
	- The placement of your cursor and the subsequent text that is selected is important when performing operations using the Payload Filter Editor. For example, you append to, replace, or change the indentation of the expression that is selected.
	- The placement of your cursor and the subsequent text that is selected is especially important when adding your Boolean operators.
	- You can include more than one varbind in the same Payload Filter expression as shown in the following example:
	`((ciaName like \Q.1.3.6.1.4.1.9.9\E.* AND ciaValue = 25) AND (ciaName like \Q.1.3.6.1.2.1.2.2.1.1.3\E.* AND ciaValue = 3))`
	In this example, a given trap must meet each of the following criteria:
	  ▪ Contain a varbind whose Object Identifier (OID) matches the regular expression `\Q.1.3.6.1.4.1.9.9\E.*` and has a value of `25`.
	  ▪ Contain a varbind whose OID matches the regular expression `\Q.1.3.6.1.2.1.2.2.1.1.3\E.*` and has a value of `3`.

Node Settings Dampen Attributes , continued

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td></td><td>

Payload Filter Editor Components

<table>
<tr><th>Attribute</th><th>Description</th></tr>
<tr><td>Attribute</td><td>

The attribute name on which NNMi searches. Filterable attributes include the following:

- ciaName
- ciaValue

</td></tr>
<tr><td>Operator</td><td>

Valid operators are described below.

- **=** Finds all values equal to the value specified. Click here for an example.

  Example: `ciaName=.1.3.6.1.4.1.9.9.13.1.2.1.7` matches any incident that contains a varbind with the name value of **.1.3.6.1.4.1.9.9.13.1.2.1.7**.

- **!=** Finds all values not equal to the value specified. Click here for an example.

  Example: `ciaName! = .1.3.6.1.4.1.9.9.13.1.2.1.7` matches any incident that contains a varbind with a name value other than **1.3.6.1.4.1.9.9.13.1.2.1.7**.

- **<** Finds all values less than the value specified. Click here for an example.

  Example: `ciaValue < 6` matches any incident that contains a varbind with a value less than **6**.

- **<=** Finds all values less than or equal to the value specified. Click here for an example.

  Example: `ciaValue <= 6` matches any incident that contains a varbind with a value less than or equal to **6**.

- **>** Finds all values greater than the value specified. Click here for an example.

  Example: `ciaValue > 4` matches any incident that contains a varbind with a value greater than **4**.

- **>=** Finds all values greater than or equal to the value specified. Click here for an example.

  Example: `ciaValue >= 4` matches any incident that contains a varbind with values greater than or equal to **4**.

- **between** Finds all values equal to and between the two values specified. Click here for an example.

  Example: `ciaValue between`

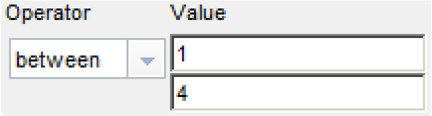
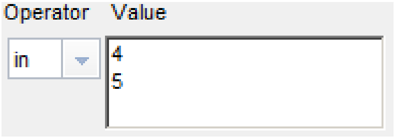
</td></tr>
</table>

</td></tr>
</table>

Node Settings Dampen Attributes , continued

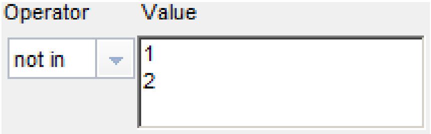
Name	Description

Payload Filter Editor Components, continued

Attribute	Description
	 Operator: between; Value: 1 / 4 matches any incident that contains a varbind value equal to or greater than **1** and equal to or less than **4**. **Note**: As shown in the example, each value must be entered on a separate line. • **in** Finds any match to at least one value in a list of values. Click here for an example. Example: `ciaValue in`  Operator: in; Value: 4 / 5 matches any incident with a varbind value of either **4** or **5**. **Note**: As shown in the example, each value must be entered on a separate line. NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example (**4, 5**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line. • **is not null** Finds all non-blank values. Click here for an example. Example: `ciaValue is not null` matches any incident with a varbind that contains a value. • **is null** Finds all blank values. Click here for an example. Example: `ciaValue is null` matches any incident with a varbind that does not contain a value. • **like** Finds matches using the syntax defined for java regular expressions. See the Pattern (Java Platform SE6) API documentation at : `http://download.oracle.com/javase/6/docs/api/java/util/regex/Pattern.html` for more information. Click here for more information. The period asterisk (.*) characters mean *any number of characters of any*

Node Settings Dampen Attributes , continued

Name	Description
	Payload Filter Editor Components, continued

Attribute	Description
	type at this location. The period (.) character means *any single character of any type at this location*. **Note**: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below. Example: `ciaName like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that begins with 1.3.6.1.4.1.9.9 and (optionally) ends with any number of characters. `ciaValue like .*Chicago.*` finds all traps or events that contain a varbind value that includes the string **Chicago**. • **not between** Finds all values except those between the two values specified. Click here for an example. Example: `ciaValue not between 5 8` matches an incident that contains a varbind with the values less than **5** or greater than **8** . • **not in** Finds all values except those included in the list of values. Click here for an example. Example: `ciaValue not in`  Operator: not in Value: 1 2 matches any incident that contains a varbind with values other than **1** and **2**. **Note**: As shown in the example, each value must be entered on a separate line. NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example, (**1, 2**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line. • **not like** Finds all that do not have the values specified using the syntax defined for Java regular expressions. See the Pattern (Java Platform SE6)

Node Settings Dampen Attributes , continued

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td></td><td>

Payload Filter Editor Components, continued

<table>
<tr><th>Attribute</th><th>Description</th></tr>
<tr><td></td><td>API documentation at: <code>http://download.oracle.com/javase/6/docs/api/java/util/regex/Pattern.html</code> for more information. Click here for an example.

The period asterisk (.*) characters mean <i>any number of characters of any type at this location</i>.

The period (.) character means <i>any single character of any type at this location</i>.

<b>Note</b>: To include literal string values in the Value attribute, enclose the string value in \Q&lt;<i>literal_value</i>&gt;\E as shown in the Examples listed below.

Example:

<code>ciaName not like \Q.1.3.6.1.4.1.9.9\E.*</code> matches any incident that contains a varbind name value that does not begin with <b>.1.3.6.1.4.1.9.9</b> and (optionally) ends with any number of characters.

<code>ciaValue not like .*Chicago.*</code> finds all traps or events that do not contain a varbind value that includes the string <b>Chicago</b>.</td></tr>
<tr><td>Value</td><td>The value for which you want NNMi to search.

Note the following:

• The values you enter are case sensitive.
• NNMi displays a variable number of value fields depending on the Operator selected. For example, the <code>between</code> Operator causes two value fields to be displayed.
• The <code>between, in</code> and <code>not in</code> operators require that each value be entered on a separate line.</td></tr>
</table>

Payload Filter Editor Buttons

<table>
<tr><th>Button</th><th>Description</th></tr>
<tr><td>Append</td><td>Appends the current expression (Attribute, Operator,and Value) to the selected expression already included in the filter string.</td></tr>
<tr><td>Insert</td><td>Inserts the current expression (Attribute, Operator,and Value) in front of the cursor location within the Filter String.</td></tr>
<tr><td>Replace</td><td>Replaces the selected expression with the expression displayed in</td></tr>
</table>

</td></tr>
</table>

Node Settings Dampen Attributes , continued

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td></td><td>
<b>Payload Filter Editor Buttons, continued</b>
<table>
<tr><th>Button</th><th>Description</th></tr>
<tr><td></td><td>the Attribute, Operator, and Value fields.</td></tr>
<tr><td>AND</td><td>Inserts the AND Boolean Operator in the selected cursor location.
<b>Note</b>: View the expression displayed under <b>Filter String</b> to see the logic of the expression as it is created.</td></tr>
<tr><td>OR</td><td>Inserts the OR Boolean Operator in the current cursor location.
<b>Note</b>: View the expression displayed under <b>Filter String</b> to see the logic of the expression as it is created.</td></tr>
<tr><td>NOT</td><td>Can be used in any part of the Filter String to specify that NNMi should exclude interfaces with values that pass the expression that immediately follows the <code>NOT</code>.
For example, when evaluating the following expression, NNMi includes interfaces with (interface description) <code>ifDesc</code> containing <b>VLAN</b>, and excludes any Interfaces that have <b>VLAN10</b> for the (interface name) <code>ifName</code> value:
<code>(ifDesc like VLAN AND NOT (ifName=VLAN10))</code>
<b>Note</b>: View the expression displayed under Filter String to see the logic of the expression as it is created .</td></tr>
<tr><td>EXISTS</td><td>Used for filters that include Capabilities or Custom Attribute names and values in the Filer String.
Indicates that you want NNMi to consider interfaces that do not have any Capabilities or Custom Attributes when evaluating the Filter String.
<b>Note</b>: If you include Capabilities or Custom Attribute names and values in the Filer String, but do not use <b>EXISTS</b> or <b>NOT EXISTS</b>, NNMi excludes from its search interfaces that do not include Capabilities or Custom Attributes.
For example, when evaluating the following Filter String, NNMi includes interfaces with (interface description) <code>ifDesc</code> containing <b>VLAN</b>, as well as any Interfaces Custom Attribute Role value is <b>LAN Connection to Oracle Server</b>:
<code>(ifDesc like VLAN OR EXISTS((customAttrName=Role AND customAttrValue=LAN Connection to Oracle Server)))</code>
<b>Note</b>: View the expression displayed under Filter String to see the</td></tr>
</table>
</td></tr>
</table>

Node Settings Dampen Attributes , continued

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td></td><td>

Payload Filter Editor Buttons, continued

<table>
<tr><th>Button</th><th>Description</th></tr>
<tr><td></td><td>logic of the expression as it is created.</td></tr>
<tr><td>NOT EXISTS</td><td>Used for filters that include Capabilities or Custom Attribute names and values in the Filer String. Indicates that you want NNMi to consider interfaces that do not have any Capabilities or Custom Attributes when evaluating the Filter String, but exclude the interfaces that match the expression that follows the <b>NOT EXISTS</b>.
<b>Note</b>: If you include Capabilities or Custom Attribute names and values in the Filer String, but do not use <b>EXISTS</b> or <b>NOT EXISTS</b>, NNMi excludes from its search interfaces that do not include Capabilities or Custom Attributes.
For example, when evaluating the following expression, NNMi includes interfaces with (interface description) <code>ifDesc</code> containing <b>VLAN</b>, and excludes any Interfaces that have the Custom Attribute Role and that Role value is <b>LAN Connection to Oracle Server</b>:
<code>(ifDesc like VLAN OR NOT EXISTS ((customAttrName=Role AND customAttrValue=LAN Connection to Oracle Server)))</code>
<b>Note</b>: View the expression displayed under Filter String to see the logic of the expression as it is created.</td></tr>
<tr><td>Delete</td><td>Deletes the selected expression.
<b>Note</b>: If the Boolean Operator is selected, the Payload Filter Editor deletes all expressions associated with the Boolean Operator.</td></tr>
</table>

</td></tr>
</table>

Configure Incident Actions for a Node Group (Management Events)

For information about each Node Settings tab:

Note: Node Settings override any other Actions settings for this incident, except those configured on the Interface Settings tab.

NNMi enables you to configure incident actions based on a Source Node's participation in a Node Group.

You can configure actions to automatically run at any point in the incident lifecycle. For example, you might want to configure an action to occur when an incident of the type you are configuring is generated (**Registered**). When an incident is generated, you might want to automatically open a trouble ticket or send email or page your network operator. After the incident is **Closed**, you might want to automatically close the trouble ticket.

> **Note:** Your actions will not be executed until you enable the Actions configuration by either clicking Enable ☑ on the Actions tab or using the **Actions** → **Enable Configuration** option.

You can configure actions for incidents generated from SNMP Trap Incidents, Syslog Messages Incidents and the NNMi Management Events Incidents. Any time an incident configuration changes, the action directory is rescanned and any Jython files are reloaded to the NNMi database. See "Lifecycle Transition Action Form (Management Events)" on page 1207 for more information about the actions directory.

Tip: Copy any required Jython files to the NNMi actions directory before you configure an incident action. New or updated actions are loaded into NNMi only when an incident configuration is updated or created. See "Lifecycle Transition Action Form (Management Events)" on page 1207 for the location of the NNMi action directory.

When the action runs, output is logged to the `incidentActions.*.*.log` file. See "Verify that NNMi Services are Running" on page 85 for more information about log files and where they are located.

To configure an automatic action for an incident:

1. Navigate to the **Management Events Configuration** tab.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Select the **Incidents** folder.
   c. Select **Management Event Configurations**.
   d. Do one of the following:
      i. To create a new incident configuration, click the ✱ New icon.
      ii. To edit an existing incident configurationselect a row, click the 📂 Open icon, and continue.
2. Select the **Node Settings** tab.
3. Do one of the following:
   a. To create a new configuration, click the ✱ New icon.
   b. To edit an existing configuration, double-click the row representing the configuration you want to edit.
4. Make sure you configure the basic Node Setting behavior. See "Configure Node Settings for a Management Event Incident" on page 1135 for more information.
5. Select the **Actions** tab.
6. From the **Lifecycle Actions** table toolbar, do one of the following:
   - To create an Action configuration, click the ✱ New icon, and continue.
   - To edit an Action configuration, double-click the row representing the configuration you want to edit, and continue.
   - To delete an Action configuration, select a row, and click the ✖ Delete icon.

7. In the "Lifecycle Transition Action Form (Management Events)" on page 1207, provide the required information.
8. Click **Save and Close** to save your changes and return to the **Management Event Configuration** form.

   The next time the lifecycle changes, NNMi launches the action associated with the lifecycle for the incident of that type.

Configure a Payload Filter for an Incident Action (Node Settings) (Management Events)

The Payload Filter Editor enables you to create expressions that further refine the filters used to select the incidents that cause the configured action to run. Make sure to design any complex Payload Filters offline as a Boolean expression first. This method can help to minimize errors when entering your expressions using the Payload Filter editor.

To create a Payload Filter expression:

1. Navigate to the **Management Events Configuration** form:
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Expand the **Incidents** folder.
   c. Select **Management Event Configurations** .
   d. Do one of the following:
      i. To create an incident configuration, click the New icon, and continue.
      ii. To edit an incident configuration, select a row, click the Open icon, and continue.
      iii. To delete an incident configuration, select a row, and click the Delete icon.
2. Select the **Node Settings** tab.
3. Do one of the following:
   a. To create a new configuration, click the New icon.
   b. To edit an existing configuration, select a row, click the Open icon, and continue.
4. Make sure you configure the basic Node Setting behavior. See "Configure Node Settings for a Management Event Incident" on page 1135 for more information.
5. Select the **Actions** tab.
6. Do one of the following:
   a. To create an Action configuration, click the New icon, and continue.
   b. To edit an Action configuration, select a row, click the Open icon, and continue.
   c. To delete an Action configuration, select a row, and click the Delete icon.
7. Make sure the Action settings are configured. See "Configure Incident Actions for a Node Group (Management Events)" on page 1163 for more information.

8. Select the **Payload Filter** tab.

9. Define your Payload Filter (see table). Also see "Guidelines for Creating a Payload Filter".

   a. Plan out the logic needed for your Filter String.

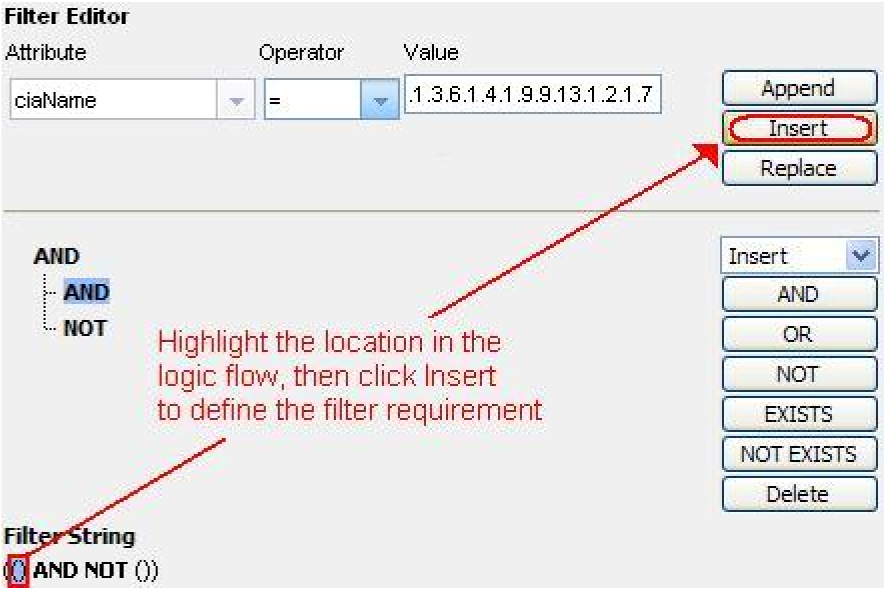
   b. Use the buttons on the bottom half of the Additional Filters Editor to establish the logic structure.

   For example, to establish the following structure, select **Insert**, then click **AND**, then **NOT**, and then **AND** a second time:

   ```
   (( ) AND NOT ( ))
   ```

   c. Now place your cursor in a location within the displayed Filter String, and use the top half of the filter editor to define the parameters of the highlighted filter requirement.

   For example, select a set of parentheses and use the Insert button to specify the filter requirement within those parentheses:



Filter Editor

Attribute: ciaName

Operator: =

Value: .1.3.6.1.4.1.9.9.13.1.2.1.7

Insert

AND
- AND
- NOT

Highlight the location in the logic flow, then click Insert to define the filter requirement

Filter String

(() AND NOT ())

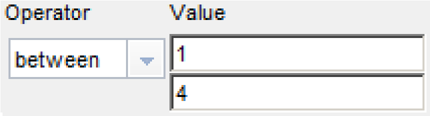
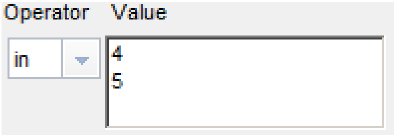
10. Click **Save and Close**.

11. Click **Save and Close** to save your changes and return to the previous form.

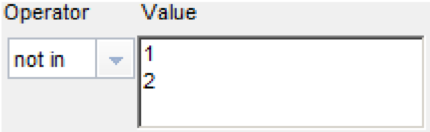
Payload Filter Editor Components

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: • ciaName • ciaValue
Operator	Valid operators are described below. • **=** Finds all values equal to the value specified. Click here for an example.

Payload Filter Editor Components, continued

Attribute	Description
	Example: `ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7` matches any incident that contains a varbind with the name value **.1.3.6.1.4.1.9.9.13.1.2.1.7**. • **!=** Finds all values not equal to the value specified. Click here for an example. Example: `ciaName != .1.3.6.1.4.1.9.9.13.1.2.1.7` matches any incident that contains a varbind with the name value other than **1.3.6.1.4.1.9.9.13.1.2.1.7**. • **<** Finds all values less than the value specified. Click here for an example. Example: `ciaValue < 6` matches any incident with a varbind value less than **6**. • **<=** Finds all values less than or equal to the value specified. Click here for an example. Example: `ciaValue <= 6` matches any incident with a varbind value less than or equal to **6**. • **>** Finds all values greater than the value specified. Click here for an example. Example: `ciaValue > 4` matches any incident with a varbind value greater than **4**. • **>=** Finds all values greater than or equal to the value specified. Click here for an example. Example: `ciaValue >= 4` matches any incident that contains a varbind with values greater than or equal to **4**. • **between** Finds all traps or events that include a varbind with a value equal to and between the two values specified. Click here for an example. Example: `ciaValue between`  Operator: between Value: 1 4 matches any incident that contains a varbind value equal to or greater than **1** and equal to or less than **4**. **Note**: As shown in the example, each value must be entered on a separate line. • **in** Finds any match to at least one value in a list of values. Click here for an example. Example: `ciaValue in`  Operator: in Value: 4 5 matches any incident with a varbind value of either **4** or **5**.

Payload Filter Editor Components, continued

Attribute	Description
	Note: As shown in the example, each value must be entered on a separate line. NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example (**4, 5**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line. • **is not null** Finds all non-blank values. Click here for an example. Example: `ciaValue is not null` matches any incident with a varbind that contains a value. • **is null** Finds all blank values. Click here for an example. Example: `ciaValue is null` matches any incident with a varbind that does not have a value. • **like** Finds matches using wildcard characters. Click here for more information about using wildcard characters. The period asterisk (.*) characters mean *any number of characters of any type at this location*. The period (.) character means *any single character of any type at this location*. **Note**: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below. Examples: `ciaName like  \Q.1.3.6.1.4.1.9.9\E.*` finds all traps or events that contain varbind names that begin with **.1.3.6.1.4.1.9.9** and (optionally) end with any number of characters. `ciaValue like .*Chicago.*` finds all traps or events that contain a varbind value that includes the string **Chicago**. • **not between** Finds all values except those between the two values specified. Click here for an example. Example: `ciaValue not between 5 8` matches an incident that contains a varbind with the values less than **5** or greater than **8**. • **not in** Finds all values except those included in the list of values. Click here for an example. Example: `ciaValue not in`  Operator: not in Value: 1 2 matches any incident that contains a varbind with values other than **1** and **2**. **Note**: As shown in the example, each value must be entered on a separate line.

Payload Filter Editor Components, continued

Attribute	Description
	NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example, (**1, 2**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line. • **not like** Finds all that do not have the values specified (using wildcard strings). Click here for an example. The period asterisk (.*) characters mean *any number of characters of any type at this location*. The period (.) character means *any single character of any type at this location*. **Note**: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below. Example: `ciaName not like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that does not begin with **.1.3.6.1.4.1.9.9**. `ciaValue not like .*Chicago.*` finds all traps or events that do not contain a varbind value that includes the string **Chicago**.
Value	The value for which you want NNMi to search. Note the following: • The values you enter are case sensitive. • NNMi displays a variable number of value fields depending on the Operator selected. For example, the `between` Operator causes two value fields to be displayed. • The `between, in` and `not in` operators require that each value be entered on a separate line.

Additional Filters Editor Buttons

Button	Description
Append	Appends the current expression (Attribute, Operator,and Value) to the selected expression already included in the Filter String.
Insert	Inserts the current expression (Attribute, Operator,and Value) in front of the cursor location within the Filter String.
Replace	Replaces the selected expression with the expression displayed in the Attribute, Operator, and Value fields.
AND	Appends, inserts, or replaces the AND Boolean Operator in the selected cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.

Additional Filters Editor Buttons, continued

Button	Description
OR	Appends, Inserts, or replaces the OR Boolean Operator in the current cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
NOT	Can be used in any part of the Filter String to specify that NNMi should exclude interfaces with values that pass the expression that immediately follows the `NOT`. For example, when evaluating the following expression, NNMi includes interfaces with (interface description) `ifDesc` containing **VLAN**, and excludes any Interfaces that have **VLAN10** for the (interface name) `ifName` value: `(ifDesc like VLAN AND NOT (ifName=VLAN10))` **Note**: View the expression displayed under Filter String to see the logic of the expression as it is created .
EXISTS	Used for filters that include Capabilities or Custom Attribute names and values in the Filer String. Indicates that you want NNMi to consider interfaces that do not have any Capabilities or Custom Attributes when evaluating the Filter String. **Note**: If you include Capabilities or Custom Attribute names and values in the Filer String, but do not use **EXISTS** or **NOT EXISTS**, NNMi excludes from its search interfaces that do not include Capabilities or Custom Attributes. For example, when evaluating the following Filter String, NNMi includes interfaces with (interface description) `ifDesc` containing **VLAN**, as well as any Interfaces Custom Attribute Role value is **LAN Connection to Oracle Server**: `(ifDesc like VLAN OR EXISTS((customAttrName=Role AND customAttrValue=LAN Connection to Oracle Server)))` **Note**: View the expression displayed under Filter String to see the logic of the expression as it is created.
NOT EXISTS	Used for filters that include Capabilities or Custom Attribute names and values in the Filer String. Indicates that you want NNMi to consider interfaces that do not have any Capabilities or Custom Attributes when evaluating the Filter String, but exclude the interfaces that match the expression that follows the **NOT EXISTS**. Note: If you include Capabilities or Custom Attribute names and values in the Filer String, but do not use **EXISTS** or **NOT EXISTS**, NNMi excludes from its search interfaces that do not include Capabilities or Custom Attributes. For example, when evaluating the following expression, NNMi includes interfaces with (interface description) `ifDesc` containing **VLAN**, and excludes any Interfaces that have the Custom Attribute Role and that Role value is **LAN Connection to Oracle Server**:

Additional Filters Editor Buttons, continued

Button	Description
	`(ifDesc like VLAN OR NOT EXISTS((customAttrName=Role AND customAttrValue=LAN Connection to Oracle Server)))` **Note**: View the expression displayed under Filter String to see the logic of the expression as it is created.
Delete	Deletes the selected expression. **Note**: If the Boolean Operator is selected, the Additional Filters Editor deletes all expressions associated with the Boolean Operator.

Configure Diagnostics Selections for a Node Group (Management Events)

For information about each Node Settings tab: .

Note: Node Settings override any other Diagnostics Selections settings for this incident, except those configured on the Interface Settings tab.

(*HP Network Node Manager iSPI Network Engineering Toolset Software*) The Diagnostic Selections form enables you to configure NNMi to automatically gather NNM iSPI NET diagnostic information for the Incident you are configuring. When using this form, you specify the diagnostics you want to run on each applicable node in the specified Node Group.

To configure Diagnostics to run on a Source Node for an incident:

1. Navigate to the **Diagnostics Selection** form.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Expand the **Incidents** folder.
   c. Select **Management Event Configurations**.
   d. Do one of the following:
      - To create an Incident configuration, click the ✳ New icon.
      - To edit an Incident configuration, select a row, click the 📂 Open icon, and continue.
   e. Navigate to **Node Settings** tab, and do one of the following:
      - To create a Node Settings configuration, click the ✳ New icon.
      - To edit a Node Settings configuration,select a row, click the 📂 Open icon, and continue.
      - To delete a Node Settings configuration, select the Node setting, and click the ✖ Delete icon.
   f. Navigate to the **Diagnostic Selection** tab, and do one of the following:

  - To create a Diagnostic Selection setting, click the New icon, and continue.
  - To edit a Diagnostic Selection setting, select a row, click the Open icon, and continue.
  - To delete a Diagnostic Selection setting, select a row, and click the Delete icon.

2. Provide the required information (see table).
3. Click **Save and Close** to save your changes and return to the previous form.

After you configure the Diagnostic for the incident and Node Group, the Diagnostic must match the following criteria before the Diagnostic runs:

- The Source Node must be in the specified Node Group.
- The Diagnostic must be valid for the Source Node. (For example, only Nortel switch Diagnostics are run on Nortel switches.)
- The incident's current lifecycle state must match a lifecycle state for which it was configured. (For example, if you configure the Incident to run a specified Diagnostic when the incident is Closed, then if the current Incident's Lifecycle State is Closed, NNMi runs that Diagnostic.)

Note: If a Source Node is in more than one Node Group, the Diagnostic is only run on the node the first time NNMi finds a match for that Source Node based on the configuration Ordering field.

If these criteria are met, NNM iSPI NET runs the Diagnostics and generates Diagnostic reports to help you solve the problem on the Source Node.

After you configure Diagnostics for an incident, you can also run Diagnostics and access the Diagnostics reports on demand, using **Actions** → **Run Diagnostics (iSPI NET only)** in the Incident form. The same criteria apply (see the criteria above). See Incident Form:Diagnostics Tab for more information.

Tip: You can right-click any object in a table or map view to access the **Actions** menu.

You can also run and access Diagnostics reports from a Node form. See Node Form: Diagnostics Tab for more information.

Diagnostic Settings Attributes

Attribute	Description
Flow Definition	Select the Diagnostic (Flow Definition) you want to use for the specified Node Group. Click the Lookup icon and choose one of the following options: • Show Analysis to display Analysis Pane information for the Flow Definition name displayed. (See Use the Analysis Pane for more information about the Analysis Pane.) • Quick Find to view the list of possible diagnostic Flow Definitions. NNMi provides diagnostics for the following types of devices: ▪ Cisco switch

Diagnostic Settings Attributes, continued

Attribute	Description
	▪ Cisco router ▪ Cisco switch/router ▪ Nortel switch See "Diagnostics (Flows) Provided by NNM iSPI NET" on page 758 for more information about the diagnostics provided and the devices to which they apply.
Lifecycle State	Incident Lifecycle State of the target Incident. If the incident's Lifecycle State matches the value specified here, the Diagnostic runs. The Diagnostic automatically runs on each applicable Source Node in the specified Node Group if the incident has the Lifecycle State currently configured in this attribute of the Diagnostic (Flow Definition - set of automated commands).
Enable	Use this attribute to temporarily disable an incident's Diagnostics settings: **Enable** □ = Temporarily disable the selected configuration. **Enable** ☑ = Enable the selected configuration.

Configure Suppression Settings for a Management Event Incident

For information about each Management Events tab:

NNMi enables you to suppress incidents based on Interface Group, Node Group, or default Suppression settings. NNMi applies your Suppression settings in the following order. Only the first match applies.

1. Interface Group (Management Event Configuration Form: Interface Settings tab)
2. Node Group (Management Event Configuration Form: Node Settings tab)
3. Suppression configuration settings without specifying an Interface Group or Node Group (Management Event Configuration Form: Suppresion tab)

A Payload Filter enables you to use the data that is included with any of the following items before they are stored as incidents in NNMi:

- Traps generated from an SNMP agent
- Syslog Messages
- Management incidents that are generated by NNMi

Examples of the type of data that can be used as a Payload Filter include SNMP trap varbind names and values as well as CIA (Custom Incident Attribute) names and values. For example, you might want NNMi to suppress a particular status change notification trap for a specified Node

Group or Interface Group. To do so, you could include the name of the trap varbind that stores this information as well as the particular status change value string the traps that you want to suppress should contain.

See "Configure Incident Suppression Settings for an Interface Group (Management Events)" on page 1099 for information about how to suppress an incident for an Interface Group with or without a Payload Filter.

See "Configure Incident Suppression Settings for a Node Group (Management Events)" on page 1136 for more information about how to suppress an incident for a Node Group with or without a Payload Filter.

To configure suppression for an incident using a Payload Filter without an Interface Group or Node Group Filter:

1. Navigate to the **Management Event Configuration** form:
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Expand the **Incidents** folder.
   c. Select **Management Event Configurations** .
   d. Do one of the following:
      i. To create an incident configuration, click the ✱ New icon, and continue.
      ii. To edit an incident configuration, select a row, click the Open icon, and continue.
      iii. To delete an incident configuration, select a row, and click the ✖ Delete icon.
2. Select the **Suppression** tab.
3. Provide the required information (see table)
4. Click **Save and Close** to save your changes and return to the previous form.

Suppression Attributes

Name	Description
Enable	Use this attribute to temporarily disable an incident's suppression settings: **Enable** □ = Temporarily disable the selected configuration. **Enable** ☑ = Enable the selected configuration.
Payload Filter	The Payload Filter Editor enables you to create expressions that further refine the filters used to select the incidents to be suppressed, enriched, or dampened. Make sure to design any complex Payload Filters offline as a Boolean expression first. This method can help to minimize errors when entering your expressions using the Payload Filter editor. When creating a Payload Filter, note the following: • Payload Filter expressions for the `like` and `not like` operators use the syntax defined for java regular expressions (java.util.regex Pattern class). • You must use a `ciaName` that already exists in the trap or event you are configuring.

Suppression Attributes , continued

Name	Description

	• Each set of expressions associated with a Boolean Operator is treated as if it were enclosed in parentheses and evaluated together.
	• View the expression displayed under **Filter String** to see the logic of the expression as it is created.
	• The `AND` and `OR` Boolean Operators must contain at least two expressions as shown in the example below.
	The following example filters incidents on voltage state:
	`AND`
	`    ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7`
	`    ciaValue = 5`
	NNMi evaluates the expression above as follows:
	`(ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7 AND ciaValue = 5)`
	NNMi finds all incidents with a varbind `.1.3.6.1.4.1.9.9.13.1.2.1.7` value of **5**.
	• The placement of your cursor and the subsequent text that is selected is important when performing operations using the Payload Filter Editor. For example, you append to, replace, or change the indentation of the expression that is selected.
	• The placement of your cursor and the subsequent text that is selected is especially important when adding your Boolean operators.
	• You can include more than one varbind in the same Payload Filter expression as shown in the following example:
	`((ciaName like \Q.1.3.6.1.4.1.9.9\E.* AND ciaValue = 25) AND (ciaName like \Q.1.3.6.1.2.1.2.2.1.1.3\E.* AND ciaValue = 3))`
	In this example, a given trap must meet each of the following criteria:
	▪ Contain a varbind whose Object Identifier (OID) matches the regular expression `\Q.1.3.6.1.4.1.9.9\E.*` and has a value of `25`.
	▪ Contain a varbind whose OID matches the regular expression `\Q.1.3.6.1.2.1.2.2.1.1.3\E.*` and has a value of `3`.

Payload Filter Editor Components

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: • ciaName

Suppression Attributes , continued

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td></td><td>

Payload Filter Editor Components, continued

<table>
<tr><th>Attribute</th><th>Description</th></tr>
<tr><td></td><td>

- ciaValue

</td></tr>
<tr><td>Operator</td><td>

Valid operators are described below.

- **=** Finds all values equal to the value specified. Click here for an example.

  Example: `ciaName=.1.3.6.1.4.1.9.9.13.1.2.1.7` matches any incident that contains a varbind with the name value of **.1.3.6.1.4.1.9.9.13.1.2.1.7**.

- **!=** Finds all values not equal to the value specified. Click here for an example.

  Example: `ciaName! = .1.3.6.1.4.1.9.9.13.1.2.1.7` matches any incident that contains a varbind with a name value other than **1.3.6.1.4.1.9.9.13.1.2.1.7**.

- **<** Finds all values less than the value specified. Click here for an example.

  Example: `ciaValue < 6` matches any incident that contains a varbind with a value less than **6**.

- **<=** Finds all values less than or equal to the value specified. Click here for an example.

  Example: `ciaValue <= 6` matches any incident that contains a varbind with a value less than or equal to **6**.

- **>** Finds all values greater than the value specified. Click here for an example.

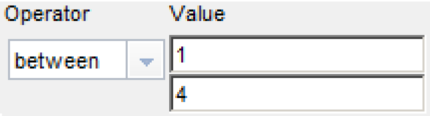
  Example: `ciaValue > 4` matches any incident that contains a varbind with a value greater than **4**.

- **>=** Finds all values greater than or equal to the value specified. Click here for an example.

  Example: `ciaValue >= 4` matches any incident that contains a varbind with values greater than or equal to **4**.

- **between** Finds all values equal to and between the two values specified. Click here for an example.

  Example: `ciaValue between`

  Operator: between
  Value: 1
  4

  matches any incident that contains a varbind value equal to or greater than

</td></tr>
</table>

</td></tr>
</table>

Suppression Attributes , continued

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td></td><td>

Payload Filter Editor Components, continued

<table>
<tr><th>Attribute</th><th>Description</th></tr>
<tr><td></td><td>

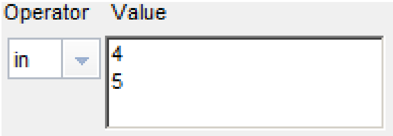
1 and equal to or less than **4**.

Note: As shown in the example, each value must be entered on a separate line.

- **in** Finds any match to at least one value in a list of values. Click here for an example.

  Example:

  `ciaValue in`

  Operator: in
  Value:
  4
  5

  matches any incident with a varbind value of either **4** or **5**.

  Note: As shown in the example, each value must be entered on a separate line.

  NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example (**4, 5**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line.

- **is not null** Finds all non-blank values. Click here for an example.

  Example: `ciaValue is not null` matches any incident with a varbind that contains a value.

- **is null** Finds all blank values. Click here for an example.

  Example: `ciaValue is null` matches any incident with a varbind that does not contain a value.

- **like** Finds matches using the syntax defined for java regular expressions. See the Pattern (Java Platform SE6) API documentation at : `http://download.oracle.com/javase/6/docs/api/java/util/regex/Pattern.html` for more information. Click here for more information.

  The period asterisk (.*) characters mean *any number of characters of any type at this location*.

  The period (.) character means *any single character of any type at this location*.

</td></tr>
</table>

</td></tr>
</table>

Suppression Attributes , continued

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td></td><td>

Payload Filter Editor Components, continued

<table>
<tr><th>Attribute</th><th>Description</th></tr>
<tr><td></td><td>

Note: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below.

Example:

`ciaName like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that begins with 1.3.6.1.4.1.9.9 and (optionally) ends with any number of characters.

`ciaValue like .*Chicago.*` finds all traps or events that contain a varbind value that includes the string **Chicago**.

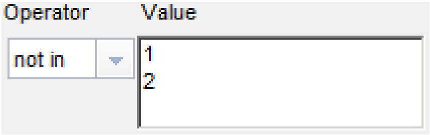
- **not between** Finds all values except those between the two values specified. Click here for an example.

  Example: `ciaValue not between 5 8` matches an incident that contains a varbind with the values less than **5** or greater than **8** .

- **not in** Finds all values except those included in the list of values. Click here for an example.

  Example:

  `ciaValue not in`

  Operator: not in
  Value:
  1
  2

  matches any incident that contains a varbind with values other than **1** and **2**.

  Note: As shown in the example, each value must be entered on a separate line.

  NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example, (**1, 2**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line.

- **not like** Finds all that do not have the values specified using the syntax defined for Java regular expressions. See the Pattern (Java Platform SE6) API documentation at: `http://download.oracle.com/javase/6/docs/api/java/util/regex/Pattern.html` for more information. Click here for an

</td></tr>
</table>

</td></tr>
</table>

Suppression Attributes , continued

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td></td><td>
<b>Payload Filter Editor Components, continued</b>
<table>
<tr><th>Attribute</th><th>Description</th></tr>
<tr><td></td><td>
example.
The period asterisk (.*) characters mean <i>any number of characters of any type at this location</i>.
The period (.) character means <i>any single character of any type at this location</i>.
<b>Note</b>: To include literal string values in the Value attribute, enclose the string value in \Q&lt;<i>literal_value</i>&gt;\E as shown in the Examples listed below.
Example:
<code>ciaName not like \Q.1.3.6.1.4.1.9.9\E.*</code> matches any incident that contains a varbind name value that does not begin with <b>.1.3.6.1.4.1.9.9</b> and (optionally) ends with any number of characters.
<code>ciaValue not like .*Chicago.*</code> finds all traps or events that do not contain a varbind value that includes the string <b>Chicago</b>.
</td></tr>
<tr><td>Value</td><td>
The value for which you want NNMi to search.
Note the following:
• The values you enter are case sensitive.
• NNMi displays a variable number of value fields depending on the Operator selected. For example, the <code>between</code> Operator causes two value fields to be displayed.
• The <code>between, in</code> and <code>not in</code> operators require that each value be entered on a separate line.
</td></tr>
</table>
<b>Payload Filter Editor Buttons</b>
<table>
<tr><th>Button</th><th>Description</th></tr>
<tr><td>Append</td><td>Appends the current expression (Attribute, Operator,and Value) to the selected expression already included in the filter string.</td></tr>
<tr><td>Insert</td><td>Inserts the current expression (Attribute, Operator,and Value) in front of the cursor location within the Filter String.</td></tr>
<tr><td>Replace</td><td>Replaces the selected expression with the expression displayed in the Attribute, Operator, and Value fields.</td></tr>
</table>
</td></tr>
</table>

Suppression Attributes , continued

Name	Description
	Payload Filter Editor Buttons, continued

Button	Description
AND	Inserts the AND Boolean Operator in the selected cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
OR	Inserts the OR Boolean Operator in the current cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
NOT	Can be used in any part of the Filter String to specify that NNMi should exclude interfaces with values that pass the expression that immediately follows the `NOT`. For example, when evaluating the following expression, NNMi includes interfaces with (interface description) `ifDesc` containing **VLAN**, and excludes any Interfaces that have **VLAN10** for the (interface name) `ifName` value: `(ifDesc like VLAN AND NOT (ifName=VLAN10))` **Note**: View the expression displayed under Filter String to see the logic of the expression as it is created .
EXISTS	Used for filters that include Capabilities or Custom Attribute names and values in the Filer String. Indicates that you want NNMi to consider interfaces that do not have any Capabilities or Custom Attributes when evaluating the Filter String. **Note**: If you include Capabilities or Custom Attribute names and values in the Filer String, but do not use **EXISTS** or **NOT EXISTS**, NNMi excludes from its search interfaces that do not include Capabilities or Custom Attributes. For example, when evaluating the following Filter String, NNMi includes interfaces with (interface description) `ifDesc` containing **VLAN**, as well as any Interfaces Custom Attribute Role value is **LAN Connection to Oracle Server**: `(ifDesc like VLAN OR EXISTS((customAttrName=Role AND customAttrValue=LAN Connection to Oracle Server)))` **Note**: View the expression displayed under Filter String to see the logic of the expression as it is created.

Suppression Attributes , continued

Name	Description
	Payload Filter Editor Buttons, continued

Button	Description
NOT EXISTS	Used for filters that include Capabilities or Custom Attribute names and values in the Filer String. Indicates that you want NNMi to consider interfaces that do not have any Capabilities or Custom Attributes when evaluating the Filter String, but exclude the interfaces that match the expression that follows the **NOT EXISTS**. **Note**: If you include Capabilities or Custom Attribute names and values in the Filer String, but do not use **EXISTS** or **NOT EXISTS**, NNMi excludes from its search interfaces that do not include Capabilities or Custom Attributes. For example, when evaluating the following expression, NNMi includes interfaces with (interface description) `ifDesc` containing **VLAN**, and excludes any Interfaces that have the Custom Attribute Role and that Role value is **LAN Connection to Oracle Server**: `(ifDesc like VLAN OR NOT EXISTS ((customAttrName=Role AND customAttrValue=LAN Connection to Oracle Server)))` **Note**: View the expression displayed under Filter String to see the logic of the expression as it is created.
Delete	Deletes the selected expression. **Note**: If the Boolean Operator is selected, the Payload Filter Editor deletes all expressions associated with the Boolean Operator.

Configure Enrichment Settings for a Management Event Incident

For information about each Management Events tab:

NNMi enables you to fine tune and enhance incidents based on Interface Group, Node Group, or default Enrichment settings. NNMi applies your Enrichment settings in the following order. Only the first match applies.

1. Interface Group (Management Event Configuration Form: Interface Settings tab)
2. Node Group (Management Event Configuration Form: Node Settings tab)
3. Enrich configuration settings without specifying an Interface Group or Node Group (Management Event Configuration Form: Enrichment tab)

The types of items you can fine tune and enhance for a selected incident configuration, include:

- Category
- Family
- Severity
- Priority
- Correlation Nature
- Message
- Assigned To

Note: Any configuration you specify for Severity, Priority, or Message overrides those values provided in the Management Event Configuration Form: Basics information.

A Payload Filter enables you to use the data that is included with any of the following items before they are stored as incidents in NNMi:

- Traps generated from an SNMP agent
- Management incidents that are generated by NNMi
- Events generated by NNM 6.x or 7.x management stations

Examples of the type of data that can be used as a Payload Filter include SNMP trap varbind names and values as well as CIA (Custom Incident Attribute) names and values. For example, you might want NNMi to suppress a particular status change notification trap for a specified Node Group or Interface Group. To do so, you could include the name of the trap varbind that stores this information as well as the particular status change value string the traps that you want to suppress should contain.

Note: The CIA added to an incident must be provided by NNMi. You cannot create CIAs.

See "Configure Incident Enrichment Settings for an Interface Group (Management Events)" on page 1107 for information about how to enrich an incident for an Interface Group with or without a Payload Filter.

See "Configure Incident Enrichment Settings for a Node Group (Management Events)" on page 1143 for more information about how to enrich an incident for a Node Group with or without a Payload Filter.

To configure Enrichment settings for an incident using a Payload Filter without an Interface Group or Node Group Filter:

1. Navigate to the **Management Event Configuration** form:
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Expand the **Incidents** folder.
   c. Select **Management Event Configurations** .
   d. Do one of the following:
      i. To create an incident configuration, click the New icon, and continue.
      ii. To edit an incident configuration, select a row, click the Open icon, and continue.

iii. To delete an incident configuration, select a row, and click the Delete icon.

2. Select the **Enrichment** tab.
3. Do one of the following:
   a. To create a new configuration, click the New icon.
   b. To edit an existing configuration, select a row, click the Open icon, and continue.
4. Provide the required information (see table)
5. Click **Save and Close** to save your changes and return to the previous form.

Enrichment Attributes

Name	Description
Category	Use the Category attribute to customize the category for this incident configuration. Possible values include: • Accounting • Application Status • Configuration • Fault • Performance • Security • Status See "Specify Category and Family Attribute Values for Organizing Your Incidents (SNMP Trap Incident)" on page 790 for more information.
Family	Use the Family attribute to customize the Family for this incident configuration. Select from the drop-down list or create a new value. For example, some of the values provided by NNMi include: • Address • Aggregated Port (Interfaces using **Link Aggregation**[1] protocol. See Interface Form: Link Aggregation tab.) • Card • Connection • Correlation

[1] Protocols used on Switches to configure multiple Interfaces (Aggregation Member Interfaces) to function as if they were one (an Aggregator Interface). When two Aggregator Interfaces establish a connection, that connection is an Aggregator Layer 2 Connection. The Aggregator Layer 2 Connection appears on Layer 2 Neighbor View maps as a thick line with an Interface icon at each end (representing the Aggregator Interface). When you double-click the thick line, it converts into multiple thin lines representing the participating Aggregation Member Layer 2 Connections with their Aggregation Member Interfaces at each end of the lines.

Enrichment Attributes , continued

Name	Description
	• Interface • Node
Severity	The incident Severity represents the seriousness calculated for the incident. Use the Severity attribute to specify the severity that should be assigned to the incident you are configuring. Possible values are described below: **Normal** - Indicates there are no known problems related to the associated object. This Severity is meant to be informational. Generally, no action is needed for these incidents. **Warning** - Indicates there might be a problem related to the associated object. **Minor** - Indicates NNMi has detected problems related to the associated object that require further investigation. **Major** - Indicates NNMi has detected problems related to the associated object to be resolved before they become critical. **Critical** - Indicates NNMi has detected problems related to the associated object that require immediate attention.
Priority	Used to communicate the urgency of resolving the selected incident. You control this value. NNMi sets this value to null by default. The lower the number the higher the priority. Possible values are: 5 **None** 4 **Low** 3 **Medium** 2 **High** 1 **Top** **Note**: The icons are displayed only in table views.
Correlation Nature	Use the Correlation Nature to customize the Correlation Nature for this incident configuration. Possible values include: • Info • None • Root Cause • Secondary Root Cause • Symptom • Stream Correlation

Enrichment Attributes , continued

Name	Description
	• Service Impact • Dedup Stream Correlation • Rate Stream Correlation See Incident Form: General Tab for more information.
Message Format	When configuring an incident, specify how the incident message appears in the incident view. The string you specify in the Message Format attribute is visible in an incident view. **Note**: The incident message limit is 1024 characters. If you exceed this limit, NNMi truncates the value starting from the right. You can use any combination of default and custom attributes: "Valid Parameters for Configuring Incident Messages (SNMP Trap Incident)" on page 795 "Include Custom Incident Attributes in Your Message Format (SNMP Trap Incident)" on page 801
Assigned To	Use to specify the owner of any incident generated for this incident configuration. Click the Lookup icon and select Quick Find to select a valid user name. **Note**: You can also begin to type a valid user name and use the auto-complete feature to select the user name of interest.
Description	Use the Description attribute to provide additional information that you want to note about the current enhancement configuration. This description applies only to the enhancement configuration and does not appear when NNMi displays any associated incident. Type a maximum of 1024 characters. Alpha-numeric and special characters (~ ! @ # $ % ^ & * ( ) _+ -) are permitted.

Configure Dampening Settings for a Management Event Incident

For information about each Management Events tab:

NNMi enables you to delay the following for an incident configuration based on the Source Object's participation in an Interface Group:

- Execution of Incident Actions
- Execution of Diagnostics (*HP Network Node Manager iSPI Network Engineering Toolset Software \ NNM iSPI NET*)
- Appearance within Incident views in the NNMi Console

You can configure the Dampening settings based on Interface Group, Node Group, or default Dampening settings. NNMi applies your Dampening settings in the following order. Only the first match applies.

1. Interface Group (Management Event Configuration Form: Interface Settings tab)
2. Node Group (Management Event Configuration Form: Node Settings tab)
3. Dampening configuration settings without specifying an Interface Group or Node Group (Management Event Configuration Form: Dampening tab)

When using the Dampening configuration, note the following:

- Duplicate and Rate Correlation incidents inherit the Dampening settings from their Correlated Children. If the Correlated Children are Closed while Dampened, and therefore deleted, NNMi retains the parent Duplicate or Rate Correlation incident. See "Correlate Duplicate Incidents (Deduplication Configuration)" on page 659 and "Track Incident Frequency (Rate: Time Period and Count)" on page 659 for more information about Duplicate and Rate Correlation incidents.

  Note: NNMi administrators can view the number of incidents Closed and deleted while dampened. Access the **Help** → **System Information** → **Health** tab, click the View Detailed Health Report button, and search for the word dampened.

- After the Dampen Interval has passed, NNMi changes the Lifecycle State to REGISTERED.

  See About the Incident Lifecycle for more information about Lifecycle State.

See "Configure Incident Dampening Settings for an Interface Group (Management Events)" on page 1119 for information about how to configure Dampening settings for an Interface Group with or without a Payload Filter.

See "Configure Incident Dampening Settings for a Node Group (Management Events)" on page 1155 for more information about how to configure Dampening settings for a Node Group with or without a Payload Filter.

To configure Dampening settings for an incident using a Payload Filter without an Interface Group or Node Group Filter:

1. Navigate to the **Management Event Configuration** form:
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Expand the **Incidents** folder.
   c. Select **Management Event Configurations**.
   d. Do one of the following:
      i. To create a configuration, click the ✱ New icon, and continue.
      ii. To edit configuration, double-click the row representing the configuration you want to edit, and continue.
      iii. To delete a configuration, select a row, and click the ✖ Delete icon.
2. Select the **Dampening** tab.
3. Provide the required information (see table)
4. Click **Save and Close** to save your changes and return to the previous form.

Dampening Attributes

Name	Description
Enable	Use this attribute to temporarily disable an incident's Dampening settings: **Enable** □ = Temporarily disable the selected configuration. **Enable** ☑ = Enable the selected configuration.
Hour	Specifies the number of hours to be used for the Dampen Interval.
Minutes	Specifies the number of minutes to be used for the Dampen Interval.
Seconds	Specifies the number of seconds to be used for the Dampen Interval.

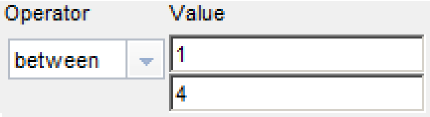
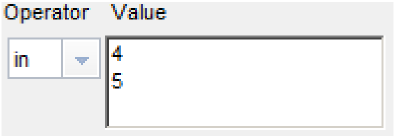
Payload Filter	The Payload Filter Editor enables you to create expressions that further refine the filters used to select the incidents to be suppressed, enriched, or dampened. Make sure to design any complex Payload Filters offline as a Boolean expression first. This method can help to minimize errors when entering your expressions using the Payload Filter editor.
	When creating a Payload Filter, note the following:
	- Payload Filter expressions for the `like` and `not like` operators use the syntax defined for java regular expressions (java.util.regex Pattern class).
	- You must use a `ciaName` that already exists in the trap or event you are configuring.
	- Each set of expressions associated with a Boolean Operator is treated as if it were enclosed in parentheses and evaluated together.
	- View the expression displayed under **Filter String** to see the logic of the expression as it is created.
	- The `AND` and `OR` Boolean Operators must contain at least two expressions as shown in the example below.
	The following example filters incidents on voltage state:
	`AND`
	`    ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7`
	`    ciaValue = 5`
	NNMi evaluates the expression above as follows:
	`(ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7 AND ciaValue = 5)`
	NNMi finds all incidents with a varbind `.1.3.6.1.4.1.9.9.13.1.2.1.7` value of **5**.
	- The placement of your cursor and the subsequent text that is selected is important when performing operations using the Payload Filter Editor. For example, you append to, replace, or change the indentation of the expression that is selected.

Dampening Attributes , continued

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td></td><td>
<ul>
<li>The placement of your cursor and the subsequent text that is selected is especially important when adding your Boolean operators.</li>
<li>You can include more than one varbind in the same Payload Filter expression as shown in the following example:
<code>((ciaName like \Q.1.3.6.1.4.1.9.9\E.* AND ciaValue = 25) AND (ciaName like \Q.1.3.6.1.2.1.2.2.1.1.3\E.* AND ciaValue = 3))</code>
In this example, a given trap must meet each of the following criteria:
<ul>
<li>Contain a varbind whose Object Identifier (OID) matches the regular expression <code>\Q.1.3.6.1.4.1.9.9\E.*</code> and has a value of <code>25</code>.</li>
<li>Contain a varbind whose OID matches the regular expression <code>\Q.1.3.6.1.2.1.2.2.1.1.3\E.*</code> and has a value of <code>3</code>.</li>
</ul>
</li>
</ul>
<b>Payload Filter Editor Components</b>
<table>
<tr><th>Attribute</th><th>Description</th></tr>
<tr><td>Attribute</td><td>The attribute name on which NNMi searches. Filterable attributes include the following:
<ul>
<li>ciaName</li>
<li>ciaValue</li>
</ul></td></tr>
<tr><td>Operator</td><td>Valid operators are described below.
<ul>
<li><b>=</b> Finds all values equal to the value specified. Click here for an example.
Example: <code>ciaName=.1.3.6.1.4.1.9.9.13.1.2.1.7</code> matches any incident that contains a varbind with the name value of <b>.1.3.6.1.4.1.9.9.13.1.2.1.7</b>.</li>
<li><b>!=</b> Finds all values not equal to the value specified. Click here for an example.
Example: <code>ciaName! = .1.3.6.1.4.1.9.9.13.1.2.1.7</code> matches any incident that contains a varbind with a name value other than <b>1.3.6.1.4.1.9.9.13.1.2.1.7</b>.</li>
<li><b>&lt;</b> Finds all values less than the value specified. Click here for an example.
Example: <code>ciaValue &lt; 6</code> matches any incident that contains a varbind with a value less than <b>6</b>.</li>
</ul></td></tr>
</table>
</td></tr>
</table>

Dampening Attributes , continued

Name	Description
	Payload Filter Editor Components, continued

Attribute	Description
	■ **<=** Finds all values less than or equal to the value specified. Click here for an example. Example: `ciaValue <= 6` matches any incident that contains a varbind with a value less than or equal to **6**. ■ **>** Finds all values greater than the value specified. Click here for an example. Example: `ciaValue > 4` matches any incident that contains a varbind with a value greater than **4**. ■ **>=** Finds all values greater than or equal to the value specified. Click here for an example. Example: `ciaValue >= 4` matches any incident that contains a varbind with values greater than or equal to **4**. ■ **between** Finds all values equal to and between the two values specified. Click here for an example. Example: `ciaValue between`  Operator: between Value: 1 4 matches any incident that contains a varbind value equal to or greater than **1** and equal to or less than **4**. **Note**: As shown in the example, each value must be entered on a separate line. ■ **in** Finds any match to at least one value in a list of values. Click here for an example. Example: `ciaValue in`  Operator: in Value: 4 5 matches any incident with a varbind value of either **4** or **5**. **Note**: As shown in the example, each value must be entered on a separate line.

Dampening Attributes , continued

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td></td><td>

Payload Filter Editor Components, continued

<table>
<tr><th>Attribute</th><th>Description</th></tr>
<tr><td></td><td>

NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example (**4, 5**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line.

- **is not null** Finds all non-blank values. Click here for an example.

  Example: `ciaValue is not null` matches any incident with a varbind that contains a value.

- **is null** Finds all blank values. Click here for an example.

  Example: `ciaValue is null` matches any incident with a varbind that does not contain a value.

- **like** Finds matches using the syntax defined for java regular expressions. See the Pattern (Java Platform SE6) API documentation at : `http://download.oracle.com/javase/6/docs/api/java/util/regex/Pattern.html` for more information. Click here for more information.

  The period asterisk (.*) characters mean *any number of characters of any type at this location*.

  The period (.) character means *any single character of any type at this location*.

  Note: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below.

  Example:

  `ciaName like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that begins with 1.3.6.1.4.1.9.9 and (optionally) ends with any number of characters.

  `ciaValue like .*Chicago.*` finds all traps or events that contain a varbind value that includes the string **Chicago**.

- **not between** Finds all values except those between the two values specified. Click here for an example.

  Example: `ciaValue not between 5 8` matches an incident that contains a varbind with the values less than **5** or greater than **8** .

</td></tr>
</table>

</td></tr>
</table>

Dampening Attributes , continued

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td></td><td>

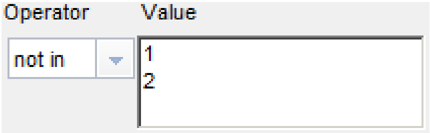
Payload Filter Editor Components, continued

<table>
<tr><th>Attribute</th><th>Description</th></tr>
<tr><td></td><td>

- **not in** Finds all values except those included in the list of values. Click here for an example.

  Example:

  `ciaValue not in`

  Operator: not in
  Value:
  1
  2

  matches any incident that contains a varbind with values other than **1** and **2**.

  Note: As shown in the example, each value must be entered on a separate line.

  NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example, (**1, 2**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line.

- **not like** Finds all that do not have the values specified using the syntax defined for Java regular expressions. See the Pattern (Java Platform SE6) API documentation at: `http://download.oracle.com/javase/6/docs/api/java/util/regex/Pattern.html` for more information. Click here for an example.

  The period asterisk (.*) characters mean *any number of characters of any type at this location*.

  The period (.) character means *any single character of any type at this location*.

  Note: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below.

  Example:

  `ciaName not like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that does not begin with **.1.3.6.1.4.1.9.9** and (optionally) ends with any number of characters.

</td></tr>
</table>

</td></tr>
</table>

Dampening Attributes , continued

Name	Description

Payload Filter Editor Components, continued

Attribute	Description
	`ciaValue not like .*Chicago.*` finds all traps or events that do not contain a varbind value that includes the string **Chicago**.
Value	The value for which you want NNMi to search. Note the following: - The values you enter are case sensitive. - NNMi displays a variable number of value fields depending on the Operator selected. For example, the `between` Operator causes two value fields to be displayed. - The `between, in` and `not in` operators require that each value be entered on a separate line.

Payload Filter Editor Buttons

Button	Description
Append	Appends the current expression (Attribute, Operator,and Value) to the selected expression already included in the filter string.
Insert	Inserts the current expression (Attribute, Operator,and Value) in front of the cursor location within the Filter String.
Replace	Replaces the selected expression with the expression displayed in the Attribute, Operator, and Value fields.
AND	Inserts the AND Boolean Operator in the selected cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
OR	Inserts the OR Boolean Operator in the current cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
NOT	Can be used in any part of the Filter String to specify that NNMi should exclude interfaces with values that pass the expression that immediately follows the `NOT`. For example, when evaluating the following expression, NNMi includes interfaces with (interface description) `ifDesc` containing

Dampening Attributes , continued

<table>
<tr><th>Nam
e</th><th>Description</th></tr>
<tr><td></td><td>

Payload Filter Editor Buttons, continued

<table>
<tr><th>Button</th><th>Description</th></tr>
<tr><td></td><td>**VLAN**, and excludes any Interfaces that have **VLAN10** for the (interface name) `ifName` value:
`(ifDesc like VLAN AND NOT (ifName=VLAN10))`
Note: View the expression displayed under Filter String to see the logic of the expression as it is created .</td></tr>
<tr><td>EXISTS</td><td>Used for filters that include Capabilities or Custom Attribute names and values in the Filer String.
Indicates that you want NNMi to consider interfaces that do not have any Capabilities or Custom Attributes when evaluating the Filter String.
Note: If you include Capabilities or Custom Attribute names and values in the Filer String, but do not use **EXISTS** or **NOT EXISTS**, NNMi excludes from its search interfaces that do not include Capabilities or Custom Attributes.
For example, when evaluating the following Filter String, NNMi includes interfaces with (interface description) `ifDesc` containing **VLAN**, as well as any Interfaces Custom Attribute Role value is **LAN Connection to Oracle Server**:
`(ifDesc like VLAN OR EXISTS ((customAttrName=Role AND customAttrValue=LAN Connection to Oracle Server)))`
Note: View the expression displayed under Filter String to see the logic of the expression as it is created.</td></tr>
<tr><td>NOT EXISTS</td><td>Used for filters that include Capabilities or Custom Attribute names and values in the Filer String. Indicates that you want NNMi to consider interfaces that do not have any Capabilities or Custom Attributes when evaluating the Filter String, but exclude the interfaces that match the expression that follows the **NOT EXISTS**.
Note: If you include Capabilities or Custom Attribute names and values in the Filer String, but do not use **EXISTS** or **NOT EXISTS**, NNMi excludes from its search interfaces that do not include Capabilities or Custom Attributes.
For example, when evaluating the following expression, NNMi</td></tr>
</table>

</td></tr>
</table>

Dampening Attributes , continued

<table>
<thead>
<tr><th>Name</th><th>Description</th></tr>
</thead>
<tbody>
<tr><td></td><td>
<p><b>Payload Filter Editor Buttons, continued</b></p>
<table>
<thead>
<tr><th>Button</th><th>Description</th></tr>
</thead>
<tbody>
<tr><td></td><td>
<p>includes interfaces with (interface description) <code>ifDesc</code> containing <b>VLAN</b>, and excludes any Interfaces that have the Custom Attribute Role and that Role value is <b>LAN Connection to Oracle Server</b>:</p>
<p><code>(ifDesc like VLAN OR NOT EXISTS ((customAttrName=Role AND customAttrValue=LAN Connection to Oracle Server)))</code></p>
<p><b>Note</b>: View the expression displayed under Filter String to see the logic of the expression as it is created.</p>
</td></tr>
<tr><td>Delete</td><td>
<p>Deletes the selected expression.</p>
<p><b>Note</b>: If the Boolean Operator is selected, the Payload Filter Editor deletes all expressions associated with the Boolean Operator.</p>
</td></tr>
</tbody>
</table>
</td></tr>
</tbody>
</table>

Configure Deduplication for a Management Event Incident

For information about each Management Events tab:

The deduplication configuration determines what values NNMi should match to detect when an SNMP trap, management event, or remote NNM 6.x/7.x event is a duplicate.

Note the following:

- Suppression, Enrichment, and Dampening are not supported for Deduplication incidents.
- NNMi applies only one deduplication configuration per incident . If NNMi generates an incident using a specified deduplication configuration, NNMi continues to correlate duplicate incidents using the original configuration. To use a different deduplication configuration for an incident, first delete the current deduplication incident (created using the original deduplication configuration). NNMi generates the next deduplication incident according to the new deduplication configuration settings.
- By default, NNMi updates the **Duplicate Count** every 30 seconds. This interval cannot be changed.
- NNMi continues to update the duplicate count regardless of an incident's lifecycle state. For example, if an incident's **Lifecycle State** is set to **Closed**, the duplicate count continues to be incremented. See About the Incident Lifecycle for more information. This behavior helps you identify situations in which the incident is not yet fixed. Take note if the Duplicate Count is

incremented after a lengthy time period has elapsed, which might indicate there is a new problem with the node, interface, or address.

- Each time you stop and restart ovjboss, any incidents that have not yet been correlated or persisted are lost. This means that after a restart of ovjboss, an incoming incident might not be correlated as expected. For example, after a restart of ovjboss, a duplicate incident might not be correlated under its original parent incident. Instead, a new parent incident might be generated. See "Stop or Start an NNMi Process" on page 82for more information about starting and stopping the ovjboss process.
- If a Duplicate Correlation Incident is dampened, note the following:
  - Duplicate Correlation Incidents inherit the Dampening settings from its Correlated Children.
  - NNMi always retains the Parent Duplicate Correlation incident, even if its Child Incidents are Closed and subsequently deleted.

    See "Dampening Incident Configurations" on page 679 for more information about Dampening an incident configuration.

To specify or delete a deduplication configuration:

1. Navigate to the **Management Event Configuration** form:
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Expand the **Incidents** folder.
   c. Select **Management Event Configurations**.
   d. Do one of the following:
      i. To create a deduplication configuration, click the ✱ New icon, and continue.
      ii. To edit a deduplication configuration, select a row, click the Open icon, and continue.
      iii. To delete a deduplication configuration, select a row, and click the ✖ Delete icon.
2. Select the **Deduplication** tab.
3. Provide the required information (see "Deduplication Attributes" table).
4. Click **Save and Close** to save your changes and return to the previous form.

Deduplication Attributes

Name	Description
Enabled	Use this attribute to temporarily disable an incident's deduplication configuration: **Enable** □ = Temporarily disable the selected configuration. **Enable** ☑ = Enable the selected configuration. **Note:** After a deduplication configuration is enabled, NNMi increments the **Duplicate Count** for an associated incident regardless of the **Lifecycle State** value. For example, if an incident's **Lifecycle State** is set to **Closed**, the duplicate count continues to be incremented. See About the Incident

Deduplication Attributes, continued

Name	Description
	Lifecycle for more information.
Count	Specifies the number of duplicate incidents for the current configuration that NNMi stores at one time. For example, if the Count is 10, after NNMi receives 10 duplicate incidents, NNMi deletes the first (oldest) duplicate incident and keeps the eleventh. (NNMi stores ten maximum.)
Hours	Used with the Minute and Second Intervals to specify the time that must elapse before a new duplicate incident is generated for this incident configuration. For example, if the Hour Interval value is 1, and no Minute or Second Intervals are specified, and the duplicate incident is not generated within one hour, NNMi generates a new duplicate incident the next time it occurs.
Minutes	Used with the Hour and Second interval to specify the time that must elapse before a new duplicate incident is generated for this incident configuration. For example, if the Minute Interval is 30 and no Hour or Second Intervals are specified, and the duplicate incident is not generated within 30 minutes, NNMi generates a new duplicate incident the next time it occurs.
Seconds	Used with the Hour and Minute Intervals to specify the time that must elapse before a new duplicate incident is generated for this incident configuration. For example, if the Second Interval is 120 and no Hour or Minute Intervals are specified, and the duplicate incident is not generated within 120 seconds, NNMi generates a new duplicate incident the next time it occurs.
Parent Incident	varUsed to specify the Incident Configuration that will be the Parent Incident for the incident you are configuring. For example, you might have created a Management Event Incident Configuration that could be used as the **Parent Incident** for SNMP Trap Incidents. When specifying the **Parent Incident**, you have the following options: • When you want to use a configuration that NNMi provides, use the default **Duplicate Correlation** incident configuration . If you select this option, the incident message for the Parent Incident begins as follows: `Duplicate Correlation for` *incident_configuration_name*> For example if you are configuring a **Node Down** incident and select **Duplicate Correlation** as the **Parent Incident**, the Parent Incident message begins with: **Duplicate Correlation for Node Down**. Each **Node Down** incident that is a duplicate then appears correlated under the **Duplicate Correlation for Node Down** incident. • NNMi also enables you to customize the Parent Incident for a given deduplication scenario. If you have created a Management Event Incident Configuration to use for this deduplication scenario, select the Management Event Incident Configuration that you have created.
Comparison	Specify the attribute values that must match before the incident is identified as a

Deduplication Attributes, continued

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td>Criteria</td><td>duplicate. The possible attributes consist of the following choices.
• **Name** - The **Name** attribute value from the Incident form: General tab.
• **CIA** - Represents any of the following items configured as a Parameter Value using the "Deduplication Comparison Parameters Form " on page 659:
▪ The **Value** attribute from the Incident form: Custom Attributes tab
▪ An SNMP varbind Object ID
▪ An SNMP varbind position number
If you want to use CIA as part of your comparison criteria, see "Deduplication Comparison Parameters Form " on page 659
• **SourceNode** - The **Source Node** attribute value from the Basics attributes listed on the Incident form. The Source Node value is the IP Address or Name of the node for which the incident was generated.
Note: The Source Node must be stored in the NNMi database.
• **Source Object** - The **Source Object** attribute value from the Basics attributes listed on the Incident form.
Note: The Source Object must be stored in the NNMi database.
Note: Each attribute value in the option you select must match before the incident is identified as a duplicate. For example, if you select **Name**, only the Incident Name value must match. If you select **Name SourceNode SourceObject CIA**, the Incident Name, Source Node, Source Object, and all Custom Incident Attribute values that you configure as a Parameter Value must match before NNMi identifies the incident as a duplicate.
Selecting an option that includes CIA enables you to further refine the deduplication criteria. For example, you might want to configure deduplication for incidents with CIA values that specify the same State attribute value for a particular network object.
For a description of each Comparison Criteria option, click here.
<table>
<tr><th>Comparison Criteria</th><th>Description</th></tr>
<tr><td>Name</td><td>Value of the **Name** attribute from the Incident form: General tab must match.</td></tr>
<tr><td>Name CIA</td><td>Each of the following values must match:
• **Name** attribute from the Incident form: General tab
• **CIA** - Represents the Value associated with any of the following items configured as a Parameter Value using the</td></tr>
</table>
</td></tr>
</table>

Deduplication Attributes, continued

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td></td><td>
<table>
<tr><th>Comparison Criteria</th><th>Description</th></tr>
<tr><td></td><td>"Deduplication Comparison Parameters Form " on page 659:
▪ Name of a Custom Incident Attribute (CIA) provided by NNMi. (See the Incident form: Custom Attributes tab
▪ An SNMP varbind Object ID
▪ An SNMP varbind position number
If you want to use CIA as part of your comparison criteria, see "Deduplication Comparison Parameters Form " on page 659</td></tr>
<tr><td>Name SourceNode</td><td>**Note**: Select this option only if the Source Node is stored in the NNMi database.
Each of the following values must match:
• **Name** attribute from the Incident form: General tab
• The **Source Node** attribute value from the Basics attributes listed on the Incident form</td></tr>
<tr><td>Name SourceNode CIA</td><td>**Note**: Select this option only if the Source Node is stored in the NNMi database.
Each of the following values must match:
• **Name** attribute from the Incident form: General tab
• The **Source Node** attribute value from the Basics attributes listed on the Incident form
• **CIA** - Represents the Value associated with any of the following items configured as a Parameter Value using the "Deduplication Comparison Parameters Form " on page 659:
▪ The **Value** attribute from the Incident form: Custom Attributes tab
▪ An SNMP varbind Object ID
▪ An SNMP varbind position number
If you want to use CIA as part of your comparison criteria, see "Deduplication Comparison Parameters Form " on page 659</td></tr>
</table>
</td></tr>
</table>

Deduplication Attributes, continued

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td></td><td>
<table>
<tr><th>Comparison Criteria</th><th>Description</th></tr>
<tr><td>Name
SourceObject</td><td>**Note**: Select this option only if the Source Object is stored in the NNMi database.
Each of the following values must match:
• **Name** attribute from the Incident form: General tab
• The **Source Object** attribute value from the Basics attributes listed on the Incident form.</td></tr>
<tr><td>Name
SourceObject
CIA</td><td>**Note**: Select this option only if the Source Object is stored in the NNMi database.
Each of the following values must match:
• **Name** attribute from the Incident form: General tab
• The **Source Object** attribute value from the Basics attributes listed on the Incident form
• **CIA** - Represents the Value associated with any of the following items configured as a Parameter Value using the "Deduplication Comparison Parameters Form " on page 659:
▪ The **Name** attribute from the Incident form: Custom Attributes tab
▪ An SNMP varbind Object ID
▪ An SNMP varbind position number
If you want to use CIA as part of your comparison criteria, see "Deduplication Comparison Parameters Form " on page 659</td></tr>
<tr><td>Name
SourceNode
SourceObject</td><td>**Note**: Select this option only if the Source Node and Source Object are stored in the NNMi database.
Each of the following values must match:
• **Name** attribute from the Incident form: General tab
• The **Source Node** attribute value from the Basics attributes listed on the Incident form
• The **Source Object** attribute value from the Basics attributes listed on the Incident form</td></tr>
<tr><td>Name</td><td>**Note**: Select this option only if the Source Node and Source</td></tr>
</table>
</td></tr>
</table>

Deduplication Attributes, continued

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td></td><td>
<table>
<tr><th>Comparison Criteria</th><th>Description</th></tr>
<tr><td>SourceNode
SourceObject
CIA</td><td>
Object are stored in the NNMi database.
<p>Each of the following values must match:</p>
<ul>
<li><b>Name</b> attribute from the Incident form: General tab</li>
<li>The <b>Source Node</b> attribute value from the Basics attributes listed on the Incident form</li>
<li>The <b>Source Object</b> attribute value from the Basics attributes listed on the Incident form</li>
<li><b>CIA</b> - Represents the Value associated with any of the following items configured as a Parameter Value using the "Deduplication Comparison Parameters Form " on page 659:
<ul>
<li>The <b>Name</b> attribute from the Incident form: Custom Attributes tab</li>
<li>An SNMP varbind Object ID</li>
<li>An SNMP varbind position number
<p>If you want to use CIA as part of your comparison criteria, see "Deduplication Comparison Parameters Form " on page 659</p></li>
</ul>
</li>
</ul>
</td></tr>
</table>
</td></tr>
<tr><td>Deduplication Comparison Parameters</td><td><i>Optional</i>. If you selected a Comparison Criteria that includes CIA, you must populate one or more rows in this table. See "Deduplication Comparison Parameters Form " on page 659.</td></tr>
</table>

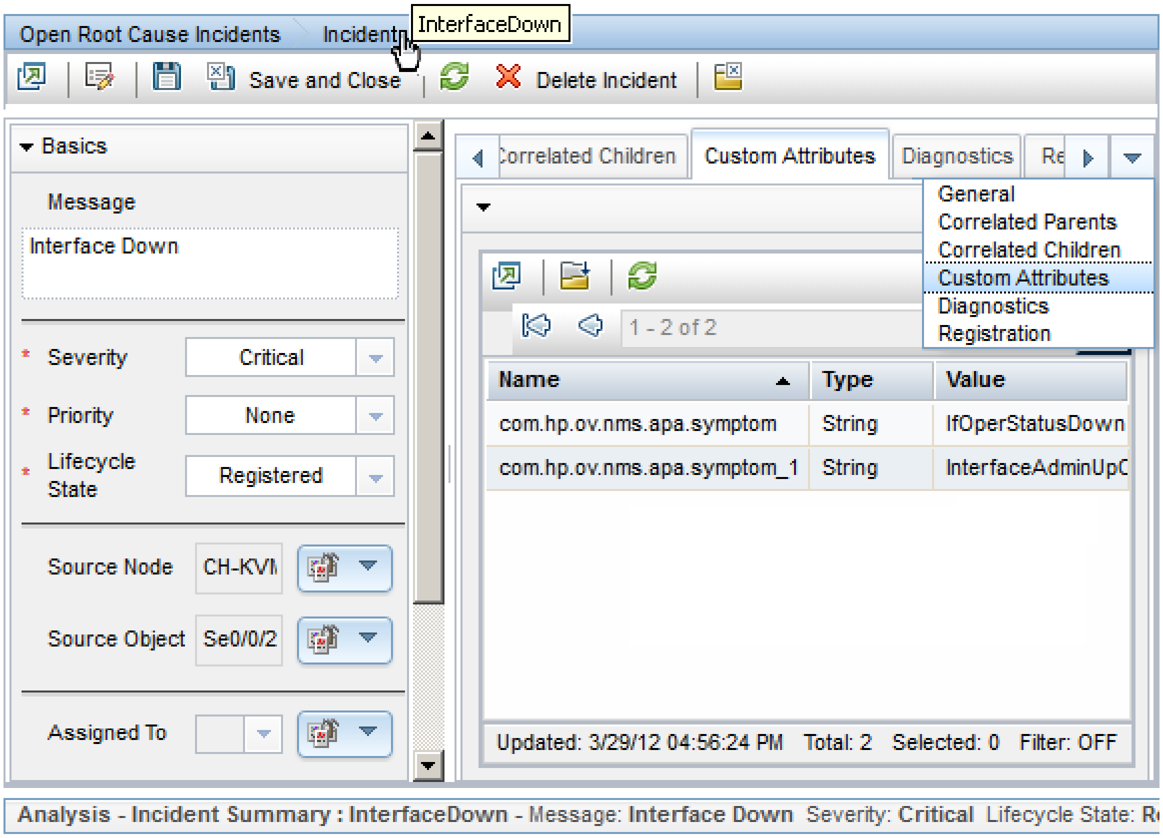
Deduplication Comparison Parameters Form (Management Events)

Comparison Parameter values enable accurate identification of duplicate incidents. Custom Incident Attributes (CIAs) are used as Comparison Parameter values. There are two categories of CIAs:

- SNMP trap varbind values (Name = the MIB varbind identifier, Type = asn_*)
- Custom attributes provided by NNMi (Name = cia.*, Type=String). See "Custom Incident Attributes Provided by NNMi (Information for Administrators)" on page 647.

The group of available CIAs depends on which incident you are configuring for this Deduplication (for example, CiscoLinkDown). To see which CIAs are available, navigate to an Incident view, select an instance of that incident-type, click the Open icon and navigate to the Custom Attributes tab. The items listed in the table are the CIAs for that particular incident-type. For example, all CiscoLinkDown incidents would have the same group of CIAs shown in the illustration below.

Note: You can also use the CIA (varbind) position number.



Message: Interface Down

Severity: Critical

Priority: None

Lifecycle State: Registered

Source Node: CH-KVI

Source Object: Se0/0/2

Assigned To:

Name	Type	Value
com.hp.ov.nms.apa.symptom	String	IfOperStatusDown
com.hp.ov.nms.apa.symptom_1	String	InterfaceAdminUpC

Updated: 3/29/12 04:56:24 PM Total: 2 Selected: 0 Filter: OFF

Analysis - Incident Summary : InterfaceDown - Message: Interface Down Severity: Critical Lifecycle State: R

To specify a CIA to use in the identification criteria for duplicate incidents:

1. Navigate to the **Deduplication Comparison Params** form.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Expand the **Incidents** folder.
   c. Select **Management Event Configurations**.
   d. Do one of the following:
      - To create a new configuration, click the ✳ New icon.
      - To edit an existing configuration, select a row, click the Open icon, and continue.
   e. On the form that opens, navigate to the **Deduplication** tab.
   f. Locate the **Deduplication Comparison Parameters** table.
   g. Do one of the following to specify which CIA:
      - To add a Custom Incident Attribute parameter specification, click the ✳ New icon.
      - To edit an existing Custom Incident Attribute parameter specification, select a row, click the Open icon, and continue.

2. In the Parameter Value field, type (or copy and paste) the exact text string from the Incident form, Custom Attribute tab, **Name** attribute value:
   - NNMi-provided CIA value (see "Custom Incident Attributes Provided by NNMi (Information for Administrators)" on page 647).
   - SNMP trap varbind identified by the Abstract Syntax Notation value (ASN.1).
3. Click **Save and Close** to save your changes and return to the previous configuration form.

Configure Rate (Time Period and Count) for a Management Event Incident

For information about each SNMP Traps tab:

Use Rate configuration to track incident patterns *based on the number of incident reoccurrences within a specified time period*. After the count within the specified time period is reached, NNMi emits a Rate Correlation incident and continues to update the Correlation Notes with the number of occurrences within that rate.

Note: Suppression, Enrichment, and Dampening are not supported for Rate incidents.

As long as your defined criteria (Count and Hours, Minutes, Seconds) is sustained, the following information is updated in the Correlation Notes of the Rate Correlation incident:

- the actual number of occurrences of incidents for that sustained rate (Count)
- the sustained time interval (Hours, Minutes, Seconds)

For example, you can set a Rate configuration to track when a link is intermittently down at least three times in 30 minutes. NNMi shows the first occurrence of the rate incident in the incident view and uses Correlation Notes to update the number of incidents and time interval to reflect all the incremental incident occurrences and time periods. To continue the example, if the rate of three times in 30 minutes is sustained for 90 minutes, NNMi updates the Correlation Notes to specify that 9 incidents occurred in 90 minutes.

NNMi provides preconfigured Rate correlations. You can add new Rate correlations.

When you open the Incident form of the newest instance:

- On the General tab, two fields notify you that the Rate correlation is working:
  - **Correlation Nature**: Rate
  - **Count**: x
- On the **Correlated Children** tab, each incident is listed in the table.
- If a Rate Correlation Incident is dampened, note the following:
  - Rate Correlation Incidents inherit the Dampening configuration settings from its Correlated Children.
  - NNMi always retains the Parent Rate Correlation Incident, even if its Child Incidents are Closed and subsequently deleted.

See "Dampening Incident Configurations" on page 679 for more information about Dampening an incident configuration.

To establish a rate correlation within an incident configuration:

1. Navigate to the **Rate** tab.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Expand the **Incidents** folder.
   c. Select **Management Event Configurations**.
   d. Do one of the following:
      - To create a new configuration, click the New icon.
      - To edit an existing configuration, select a row, click the Open icon, and continue.
   e. On the form that opens, locate the **Rate** tab.
2. Provide the definition for this Rate Configuration (see the "Rate Configuration Definition" table).
3. *Optional*. If your Comparison Criteria includes custom incident attributes (CIA) to identify one specific incident, use the Comparison Parameter List table to define each CIA.See "Rate Comparison Parameters Form" on page 678.
4. Click **Save and Close** to save your changes and return to the previous form.

Rate Configuration Definition

Attribute	Description
Enable	Use this attribute to temporarily disable an incident's rate settings:If enabled, NNMi actively tracks any reoccurrences of the designated incident within the time period you specify, and generates a Rate incident. **Enable** □ = Temporarily disable the selected configuration. **Enable** ☑ = Enable the selected configuration.
Count	Specify the number of reoccurrences required before your Rate Configuration starts working.
Hours	Used with the Minutes and Seconds attributes to specify the time duration within which the reoccurrences are measured.
Minutes	Used with the Hours and Seconds attributes to specify the time duration within which the reoccurrences are measured.
Seconds	Used with the Hours and Minutes attributes to specify the time duration within which the reoccurrences are measured.
Parent Incident	Click the icon and select Quick Find. Select **Rate Correlation** from the list.
Comparison	Specify which group of attributes must match before the incident is identified as a

Rate Configuration Definition , continued

Attribute	Description
Criteria	duplicate. The possible groups of attributes consist of the following choices. **Name** value of the Incident (from the General tab on the Incident form). **Source Node** value (from the Basics group on the Incident form). Address or name of the node for which the incident was generated. **Source Object** value (from the Basics group on the Incident form). For example, the Source Object for a LinkDown incident is **interface**. **CIA** custom incident attribute values (select from the list displayed on the Custom Attributes tab on the Incident form). If you want to use CIA as part of your comparison criteria, see "Rate Comparison Parameters Form (Management Events)" below.
Rate Comparison Parameters	*Optional*. If you selected a Comparison Criteria that includes CIA, you must populate one or more rows in this table. See "Rate Comparison Parameters Form (Management Events)" below.

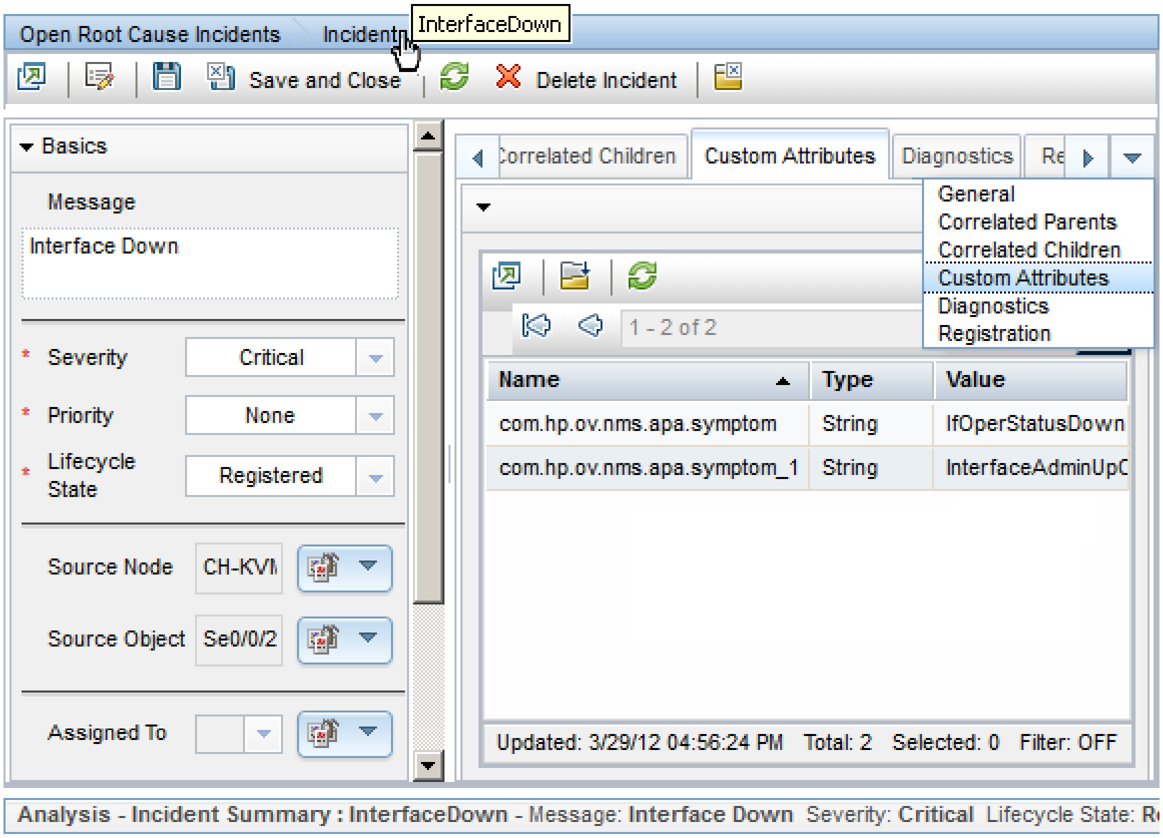
Rate Comparison Parameters Form (Management Events)

Custom Incident Attributes (CIAs) are used as parameter values. Parameter values enable accurate identification of duplicate incidents. There are two categories of CIAs:

- SNMP trap varbind values (Name = the MIB varbind identifier, Type = asn_*)
- Custom attributes provided by NNMi (Name = cia.*, Type=String). See "Custom Incident Attributes Provided by NNMi (Information for Administrators)" on page 647.

The group of available CIAs depends on which incident you are configuring for this Rate (for example, CiscoLinkDown). To see which CIAs are available, navigate to an Incident view, select an instance of that incident-type, click the Open icon and navigate to the Custom Attributes tab. The items listed in the table are the CIAs for that particular incident-type. For example, all CiscoLinkDown incidents would have the same group of CIAs shown in the illustration below.

Note: You can also use the CIA (varbind) position number.



Message	Interface Down
* Severity	Critical
* Priority	None
* Lifecycle State	Registered
Source Node	CH-KVI
Source Object	Se0/0/2
Assigned To	

Name	Type	Value
com.hp.ov.nms.apa.symptom	String	IfOperStatusDown
com.hp.ov.nms.apa.symptom_1	String	InterfaceAdminUpC

Updated: 3/29/12 04:56:24 PM Total: 2 Selected: 0 Filter: OFF

Analysis - Incident Summary : InterfaceDown - Message: Interface Down Severity: Critical Lifecycle State: R

To specify a CIA to use in the identification criteria for duplicate incidents:

1. Navigate to the **Rate Comparison Params** form.

   a. From the workspace navigation panel, select the **Configuration** workspace.

   b. Expand the **Incidents** folder.

   c. Select **Management Event Configurations** .

   d. Do one of the following:

      - To create a new configuration, click the New icon.

      - To edit an existing configuration, select a row, click the Open icon, and continue.

   e. On the form that opens, navigate to the **Rate** tab.

   f. Locate the **Rate Comparison Parameters** table.

   g. Do one of the following to specify which CIA:

      - To add a Custom Incident Attribute parameter specification, click the New icon.

      - To edit an existing Custom Incident Attribute parameter specification, select a row, click the Open icon, and continue.

2. In the Parameter Value field, type (or copy and paste) the exact text string from the Incident form, Custom Attribute tab, **Name** attribute value:

- NNMi-provided CIA value (see "Custom Incident Attributes Provided by NNMi (Information for Administrators)" on page 647).
- SNMP trap varbind identified by the Abstract Syntax Notation value (ASN.1).

3. Click **Save and Close** to save your changes and return to the previous configuration form.

Configure Actions for a Management Event Incident

For information about each Management Events tab:

For information about each Actions tab:

You can configure actions to automatically run at any point in the incident lifecycle. For example, you might want to configure an action to occur when an incident of the type you are configuring is generated (**Registered**). When an incident is generated, you might want to automatically open a trouble ticket or send email or page your network operator. After the incident is **Closed**, you might want to automatically close the trouble ticket.

> **Note:** Your actions will not be executed until you enable the Actions configuration by either clicking Enable ☑ on the Actions tab or using the **Actions** → **Enable Configuration** option.

> **Note:** If the NNMi management server is running on a Windows operating system, NNMi runs each action that you configure using the Local System account. If the NNMi management server is running on an HP-UX, Solaris or Linux operating system, NNMi runs each action that you configure using the bin user name. To change the user account associated with actions, see the "Setting the Action Server Name Parameter" section in the *HP Network Node Manager i Software Deployment Reference*.

You can configure actions for incidents generated from SNMP traps, NNM 6.x or 7.x events, and the NNMi management events. Any time an incident configuration changes, the action directory is rescanned and any Jython files are reloaded to the NNMi database. See "Lifecycle Transition Action Form (Management Events)" on the next page for more information about the actions directory.

Tip: Copy any required Jython files to the NNMi actions directory before you configure an incident action. New or updated actions are loaded into NNMi only when an incident configuration is updated or created. See "Lifecycle Transition Action Form (Management Events)" on the next page for the location of the NNMi action directory.

When the action runs, output is logged to the `incidentActions.*.*.log` file. To view the contents of the Actions log, use the **Tools** → **Incident Actions Log** menu option.

See "Verify that NNMi Services are Running" on page 85 for more information about log files and where they are located.

NNMi sets the default values described in the following table.

See the "Maintaining NNMi" chapter in the *HP Network Node Manager i Software Deployment Reference* for information about changing the default values for Action Server Properties.

Action Server Properties

Property	Description	Value
numProcess	Number of actions that can be run at one time.	10
numJythonThreads	Number of threads the action server uses to run Jython scripts.	10
userName	User name under which the action server runs.	bin

To configure an automatic action for an incident:

1. Navigate to the **Actions** tab.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Expand the **Incidents** folder.
   c. Select **Management Event Configurations**.
   d. Do one of the following:
      - To create an incident configuration, click the New icon, and continue.
      - To edit an incident configuration, select a row, click the Open icon, and continue.
      - To delete an incident configuration, select a row, and click the Delete icon.
   e. Select the **Actions** tab.
2. From the **Lifecycle Actions** table toolbar, do one of the following:
   - To create an Action configuration, click the New icon, and continue.
   - To edit an Action configuration, select a row, click the Open icon, and continue.
   - To delete an Action configuration, select a row, and click the Delete icon.
3. In the "Lifecycle Transition Action Form (Management Events)" below, provide the required information.
4. Click **Save and Close** to save your changes and return to the previous form.

   The next time the lifecycle changes, NNMi launches the action associated with the lifecycle for the incident of that type.

Lifecycle Transition Action Form (Management Events)

For information about each Actions tab:

Use this form to enter the command you want to run when an incident of the type you are configuring is at a particular Lifecycle State. For example, when an incident is generated (**Registered**), you might want to automatically open a trouble ticket or email or page your network operator.

Note: Your actions will not be executed until you enable the Actions configuration by either clicking Enable ☑ on the Actions tab or using the **Actions** → **Enable Configuration** option.

To configure an action for an incidents:

1. Navigate to the **Lifecycle Transition Actions** form:
   a. From the workspace navigation pane, select the **Configuration** workspace.
   b. Expand the **Incidents** folder.
   c. Select **Management Event Configurations**.
   d. Select the **Actions** tab.
   e. From the **Lifecycle Transition Action** table toolbar, do one of the following:
      - To create an Action configuration, click the New icon, and continue.
      - To edit an Action configuration, select a row, click the Open icon, and continue.
      - To delete an Action configuration, select a row, and click the Delete icon.
2. Make your configuration choices (see table).

   Note: NNMi reloads the configuration information anytime the incident configuration is changed.
3. Click **Save and Close** to save your changes and return to the previous form.

Create Action Attributes

Attribute	Description
Lifecycle State	Select a Lifecycle State from the drop-down menu.
Command Type	If you provided a Jython command, select **Jython** from the drop-down list. If you are using an executable or bat file, select **ScriptOrExecutable** from the drop-down list.
Command	Enter one of the following: • A Jython method with the required parameters. • Executable command for the current operating system with the required parameters. When entering a **Command** value, note the following: • Left or right bracket ([ ]) and backtick (` Unicode character: 0060 hex = 96 dec) characters are not permitted in the **Command** attribute. If you need these characters in your shell script, place them in a shell script file and reference that file from the **Command** attribute. • **Windows only**: Shell commands are not permitted in the **Command** attribute. To use shell commands, place them in a shell script file and reference that file from the **Command** attribute. • Use absolute paths to executables instead of relying on the PATH variable as it might not be set correctly. • Verify that you do not have two Jython methods with the same name. Otherwise, NNMi is not able to tell which is the correct method to load.

Create Action Attributes, continued

Attribute	Description

	- You can use the same Jython method for more than one incident configuration.
	- Jython (.py) files must reside in the following directory:
	Note: All the functions defined in the Jython files that reside in this directory are also accessible by NNMi. The files are also executed by NNMi on startup.
	Windows:
	`%NnmDataDir%\shared\nnm\actions`
	UNIX:
	`/var/opt/OV/shared/nnm/actions`
	- When using executable files, specify the absolute path to the executable command or make sure the directory in which the executable file resides is in your PATH environment variable.
	- NNMi provides a set of parameters that can pass attribute values from an incident configuration. See "Valid Parameters for Configuring Incident Actions (Management Events)" on page 1216 for more information.

Configure a Payload Filter for an Action (Management Events)

The Payload Filter Editor enables you to create expressions that further refine the filters used to select the incidents that cause the configured action to run. Make sure to design any complex Payload Filters offline as a Boolean expression first. This method can help to minimize errors when entering your expressions using the Payload Filter editor.

To create a Payload Filter expression:

1. Navigate to the **Management Event Configuration** form:
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Expand the **Incidents** folder.
   c. Select **Management Event Configurations**.
   d. Do one of the following:
      i. To create an incident configuration, click the ✳ New icon, and continue.
      ii. To edit an incident configuration, select a row, click the 📂 Open icon, and continue.
      iii. To delete an incident configuration, select a row, and click the ✖ Delete icon.
2. Select the **Actions** tab.
3. Do one of the following:

   a. To create a new configuration, click the * New icon.

   b. To edit an existing configuration, select a row, click the Open icon, and continue.

4. Select the **Payload Filter** tab.

5. Define your Payload Filter (see table). Also see "Guidelines for Creating a Payload Filter".

   a. Plan out the logic needed for your Filter String.

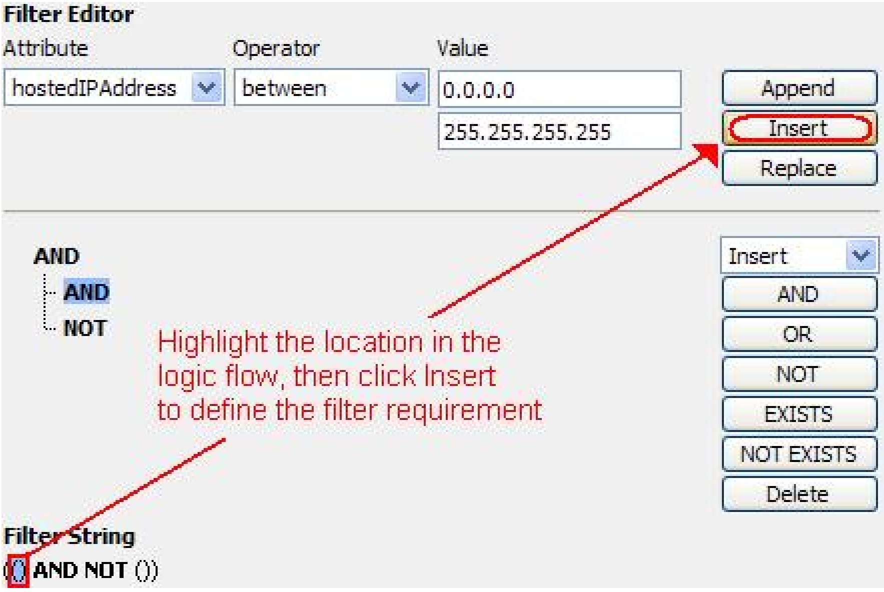
   b. Use the buttons on the bottom half of the Additional Filters Editor to establish the logic structure.

      For example, to establish the following structure, select **Insert**, then click **AND**, then **NOT**, and then **AND** a second time:

      ```
      (( ) AND NOT ( ))
      ```

   c. Now place your cursor in a location within the displayed Filter String, and use the top half of the filter editor to define the parameters of the highlighted filter requirement.

      For example, select a set of parentheses and use the Insert button to specify the filter requirement within those parentheses:



Filter Editor

Attribute	Operator	Value
hostedIPAddress	between	0.0.0.0
		255.255.255.255

Filter String

(() AND NOT ())

Highlight the location in the logic flow, then click Insert to define the filter requirement

6. Click **Save and Close**.

7. Click **Save and Close** to save your changes and return to the previous form.

When creating a Payload Filter, note the following:

- Payload Filter expressions for the `like` and `not like` operators use the syntax defined for java regular expressions (java.util.regex Pattern class)
- You must use a `ciaName` that already exists in the trap or event you are configuring.
- Each set of expressions associated with a Boolean Operator is treated as if it were enclosed in parentheses and evaluated together.

- View the expression displayed under **Filter String** to see the logic of the expression as it is created.
- The `AND` and `OR` Boolean Operators must contain at least two expressions as shown in the example below.

  The following example filters incidents on voltage state. Using this Payload Filter, you could then configure the Basics settings of the Enrichment Configuration to set the severity and message format to all incidents that return a state value of `4` or `5`.

  ```
  OR
    ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7
     ciaValue = 4
    AND
     ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7
     ciaValue = 5
  ```

  NNMi evaluates the expression above as follows:

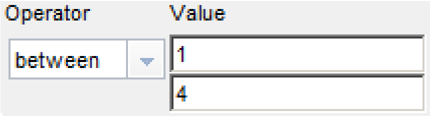
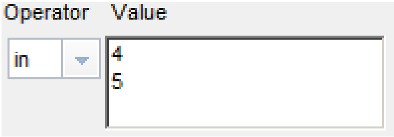
  ```
  (ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7 AND ciaValue = 4) OR (ciaName
  = .1.3.6.1.4.1.9.9.13.1.2.1.7 AND ciaValue = 5)
  ```

  NNMi finds all incidents with a varbind value of `.1.3.6.1.4.1.9.9.13.1.2.1.7` and CIA value of **4** or **5**.
- The placement of your cursor and the subsequent text that is selected is important when performing operations using the Payload Filter Editor. For example, you append to, replace, or change the indentation of the expression that is selected.
- The placement of your cursor and the subsequent text that is selected is especially important when adding your Boolean operators.

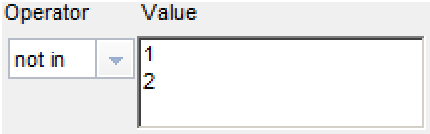
Payload Filter Editor Components

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: • ciaName • ciaValue
Operator	Valid operators are described below. • **=** Finds all values equal to the value specified. Click here for an example. Example: `ciaName=.1.3.6.1.4.1.9.9.13.1.2.1.7` matches any incident that contains a varbind with the name value **.1.3.6.1.4.1.9.9.13.1.2.1.7**. • **!=** Finds all values not equal to the value specified. Click here for an example. Example: `ciaName!=.1.3.6.1.4.1.9.9.13.1.2.1.7` matches any incident that contains a varbind with the name value other than **1.3.6.1.4.1.9.9.13.1.2.1.7**.

Payload Filter Editor Components, continued

Attribute	Description
	• **<** Finds all values less than the value specified. Click here for an example. Example: `ciaValue < 6` matches any incident that contains a varbind value less than **6**. • **<=** Finds all values less than or equal to the value specified. Click here for an example. Example: `ciaValue <= 6` matches any incident that contains a varbind value less than or equal to **6**. • **>** Finds all values greater than the value specified. Click here for an example. Example: `ciaValue > 4` matches any incident that contains a varbind value greater than **4**. • **>=** Finds all values greater than or equal to the value specified. Click here for an example. Example: `ciaValue >= 4` matches any incident that contains a varbind with values greater than or equal to **4**. • **between** Finds all traps or events that include a varbind value equal to and between the two values specified. Click here for an example. Example: `ciaValue between`  Operator between Value 1 4 matches any incident that contains a varbind value equal to or greater than **1** and equal to or less than **4**. **Note**: As shown in the example, each value must be entered on a separate line. • **in** Finds any match to at least one value in a list of values. Click here for an example. Example: `ciaValue in`  Operator in Value 4 5 matches any incident that contains a varbind value of either **4** or **5**. **Note**: As shown in the example, each value must be entered on a separate line. NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example (**4, 5**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line.

Payload Filter Editor Components, continued

Attribute	Description
	• **is not null** Finds all non-blank values. Click here for an example. Example: `ciaValue is not null` matches any incident with varbind values. • **is null** Finds all blank values. Click here for an example. Example: `ciaValue is null` matches any incident with no varbind values. • **like** Finds matches using wildcard characters. Click here for more information about using wildcard characters. The period asterisk (.*) characters mean *any number of characters of any type at this location*. The period (.) character means *any single character of any type at this location*. **Note**: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below. Examples: `ciaName like  \Q .1.3.6.1.4.1.9.9\E.*` finds all traps or events that contain varbind names that begin with **.1.3.6.1.4.1.9.9** and (optionally) end with any number of characters. `ciaValue like .*Chicago.*` finds all traps or events that contain a varbind value that includes the string **Chicago**. • **not between** Finds all values except those between the two values specified. Click here for an example. Example: `ciaValue not between 5 8` matches an incident that contains a varbind with the values less than **5** or greater than **8**. • **not in** Finds all values except those included in the list of values. Click here for an example. Example: `ciaValue not in`  Operator: not in Value: 1 2 matches any incident that contains a varbind with values other than **1** and **2**. **Note**: As shown in the example, each value must be entered on a separate line. NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example, (**1, 2**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line. • **not like** Finds all that do not have the values specified (using wildcard strings). Click here for an example.

Payload Filter Editor Components, continued

Attribute	Description
	The period asterisk (.*) characters mean *any number of characters of any type at this location*. The period (.) character means *any single character of any type at this location*. **Note**: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below. Example: `ciaName not like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that does not begin with **`.1.3.6.1.4.1.9.9`**. `ciaValue not like .*Chicago.*` finds all traps or events that do not contain a varbind value that includes the string **Chicago**.
Value	The value for which you want NNMi to search. Note the following: • The values you enter are case sensitive. • NNMi displays a variable number of value fields depending on the Operator selected. For example, the `between` Operator causes two value fields to be displayed. • The `between, in` and `not in` operators require that each value be entered on a separate line.

Payload Filter Editor Buttons

Button	Description
Append	Appends the current expression (Attribute, Operator,and Value) to the selected expression already included in the filter string.
Insert	Inserts the current expression (Attribute, Operator,and Value) in front of the cursor location within the Filter String.
Replace	Replaces the selected expression with the expression displayed in the Attribute, Operator, and Value fields.
AND	Inserts the AND Boolean Operator in the selected cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
OR	Inserts the OR Boolean Operator in the current cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
NOT	Can be used in any part of the Filter String to specify that NNMi should exclude interfaces with values that pass the expression that immediately follows the `NOT`.

Payload Filter Editor Buttons, continued

Button	Description
	For example, when evaluating the following expression, NNMi includes interfaces with (interface description) `ifDesc` containing **VLAN**, and excludes any Interfaces that have **VLAN10** for the (interface name) `ifName` value: `(ifDesc like VLAN AND NOT (ifName=VLAN10))` **Note**: View the expression displayed under Filter String to see the logic of the expression as it is created.
EXISTS	Used for filters that include Capabilities or Custom Attribute names and values in the Filer String. Indicates that you want NNMi to consider interfaces that do not have any Capabilities or Custom Attributes when evaluating the Filter String. **Note**: If you include Capabilities or Custom Attribute names and values in the Filer String, but do not use **EXISTS** or **NOT EXISTS**, NNMi excludes from its search interfaces that do not include Capabilities or Custom Attributes. For example, when evaluating the following Filter String, NNMi includes interfaces with (interface description) `ifDesc` containing **VLAN**, as well as any Interfaces Custom Attribute Role value is **LAN Connection to Oracle Server**: `(ifDesc like VLAN OR EXISTS((customAttrName=Role AND customAttrValue=LAN Connection to Oracle Server)))` **Note**: View the expression displayed under Filter String to see the logic of the expression as it is created.
NOT EXISTS	Used for filters that include Capabilities or Custom Attribute names and values in the Filer String. Indicates that you want NNMi to consider interfaces that do not have any Capabilities or Custom Attributes when evaluating the Filter String, but exclude the interfaces that match the expression that follows the **NOT EXISTS**. Note: If you include Capabilities or Custom Attribute names and values in the Filer String, but do not use **EXISTS** or **NOT EXISTS**, NNMi excludes from its search interfaces that do not include Capabilities or Custom Attributes. For example, when evaluating the following expression, NNMi includes interfaces with (interface description) `ifDesc` containing **VLAN**, and excludes any Interfaces that have the Custom Attribute Role and that Role value is **LAN Connection to Oracle Server**: `(ifDesc like VLAN OR NOT EXISTS((customAttrName=Role AND customAttrValue=LAN Connection to Oracle Server)))` **Note**: View the expression displayed under Filter String to see the logic of the expression as it is created.
Delete	Deletes the selected expression. **Note**: If the Boolean Operator is selected, the Payload Filter Editor deletes all expressions associated with the Boolean Operator.

Valid Parameters for Configuring Incident Actions (Management Events)

When configuring incident actions, consider using incident information as part of the action. NNMi provides the following parameter values. Use these parameters as variables in your Jython or executable files.

Tip: See the Using the Incident Form for more information about the parameter values.

Note: NNMi stores varbind values as custom incident attributes (CIAs).

Tip: If a value is not stored for a parameter, it is returned as “null”.

See "Lifecycle Transition Action Form" on page 748 for more information about configuring incident actions.

Valid Parameters Visible From an Incident's Form

Parameter Value	Description
$category, $cat	Value of the Category attribute in the Incident form.
$count, $cnt	Value representing the number of Custom Incident Attributes that appear in the Incident form.
$family, $fam	Value from the Family attribute in the Incident form.
$firstOccurrenceTime, $fot	Value from the First Occurrence Time attribute in the incident form.
$lastOccurrenceTime, $lot	Value from the Last Occurrence Time attribute in the incident form.
$lifecycleState, $lcs	Value from the Lifecycle State attribute in the Incident form.
$name	Value of the Name attribute from the incident configuration.
$nature, $nat	Value from the Nature attribute in the Incident form.
$origin, $ori	Value from the Origin attribute in the Incident form.
$originOccurrenceTime, $oot	Value from the Origin Occurrence Time attribute in the incident form.
$priority, $pri	Value from the Priority attribute in the Incident form.
$severity, $sev	Value of the Severity attribute of the Incident form.

Valid Parameters Visible from a Node Form

Parameter Value	Description
$managementAddress, $mga	Value from the Management Address attribute

Valid Parameters Visible from a Node Form, continued

	of the incident's source Node's form or SNMP Agent form.
$otherSideOfConnectionManagementAddress, $oma	If the incident's Source Node is part of a Layer 2 Connection, this attribute is the value of the Management Address of a node on the other side of the Layer 2 Connection.
$sourceNodeLongName, $sln	The fully-qualified DNS name as displayed in the Hostname attribute of the incident's source Node's form.
$sourceNodeName, $snn	Value from the Name attribute of the incident's source Node's form.
$sysContact, $sct	Value from the System Contact attribute of the incident's source Node form: General tab.
$sysLocation, $slc	Value from the System Location attribute of the incident's source Node form: General tab.

Valid Parameters Visible from an Interface Form

Parameter Value	Description
$ifAlias, $ifa	Value from the IfAlias attribute for the interface that is the incident's source object.
$ifConfigDupSetting, $icd	Configured Duplex Setting on the port associated with the interface that is the incident's source object.
$ifDesc, $idc	Value from the ifDesc attribute for the interface that is the incident's source object.
$ifIndex, $idx	Value from the ifIndex attribute for the interface that is the incident's source object.
$ifIpAddr, $iia	IP Address values associated with the interface that is the incident's source object. If multiple IPaddresses are associated with the interface, this parameter returns a comma-separated list.
$ifName, $ifn	Value from the ifName attribute for the interface that is the incident's source object.
$ifPhysAddr, $ipa	Value from the Physical Address attribute for the interface that is the incident's source object.
$ifSpeed, $isp	Value from the ifSpeed attribute for the interface that is the incident's souce object.
$ifType, $itp	Value from the ifType attribute for the interface that is the incident's souce object.

Valid Parameters Visible from a Layer 2 Connection Form

Parameter Value	Description
$otherSideOfConnectionConfigDupSetting, $ocd	If the incident's source Node is part of a Layer 2 Connection, this parameter contains the Configured Duplex Setting on the port associated with the interface on the other side of the connection.
$otherSideOfConnectionIfAlias, $oia	If the incident's Source Node is part of a Layer 2 Connection, this parameter is the value of the ifAlias of one of the interfaces on the other side of the Layer 2 Connection.
$otherSideOfConnectionIfDesc, $odc	If the incident's Source Node is part of a Layer 2 Connection, this parameter contains the ifDescr attribute value for the interface on the other side of the Layer 2 Connection.
$otherSideOfConnectionIfIndex, $odx	If the incident's Source Node is part of a Layer 2 Connection, this parameter contains the ifIndex attribute value for the interface on the other side of the connection.
$otherSideOfConnectionIfName, $ofn	If the incident's Source Node is part of a Layer 2 Connection, this parameter contains the ifName attribute value for the interface on the other side of the connection.

Valid Parameters Visible from a VLAN Form

Parameter Value	Description
$impVlanIds, $ivi	Value from the VLAN Id attribute associated with the interface that is the incident's source object. To access this information from an interface form, navigate to the VLAN Port tab and open the form for the VLAN of interest. If the interface is part of more than one VLAN, this parameter returns a comma-separated list.
$impVlanNames, $ivn	Value from the Global VLAN Name attribute associated with the interface that is the incident's source object. To access this information from a Node form or Interface form, navigate to the VLAN Ports tab. If the node or interface is part of more than one VLAN, this parameter returns a comma-separated list.

Valid Parameters Not Visible From a Form

Parameter Value	Description
$id	Unique Object Identifier attribute value for the incident (unique across the entire NNMi Database).
$firstOccurrenceTimeMs, $fms	Value from the First Occurrence Time attribute in the incident form, converted to millseconds (measured since January 1, 1970, 00:00:00 GMT - Greenwich Mean Time).
$lastOccurrenceTimeMs, $lms	Value from the Last Occurrence Time attribute in the incident form, converted to millseconds (measured since January 1, 1970, 00:00:00 GMT - Greenwich Mean Time).
$messageFormat, $msg	*Valid for Incident actions only*. Message text displayed for an incident when this parameter is included as an argument to an incident action.
$oid	Value of the unique object identifier (oid) for the incident configuration that originated from either an SNMP trap, Management Event, or Remote NNM 6.x or 7.x event.
$otherSideOfConnection, $osc	If the incident's Source Node is part of a Layer 2 Connection, this attribute is the following combination of values for the node and one of its interfaces on the other side of the Layer 2 Connection: The fully-qualified DNS name of the node appended with the interface Name in the following format: <*fully-qualified DNS name*> [*interface_name*]
$originOccurrenceTimeMs, $oms	Value from the Origin Occurrence Time attribute in the incident form, converted to millseconds (measured since January 1, 1970, 00:00:00 GMT - Greenwich Mean Time).
$sourceNodeUuid, $snu	Universally Unique Object Identifier attribute value of the source node object for the incident (unique across all databases). This identifier distinguishes the source node object instance from all other node objects.
$sourceObjectClass, $soc	Value of the object class for the object you want to include. Use this parameter to request more details of a class of objects through a web service. Examples of object classes include: `com.hp.ov.nms.model.core.Interface` and `com.hp.ov.nms.model.snmp.SnmpAgent.`
$sourceObjectName, $son	Value from the Name attribute of the source object. For example, an interface object is named according to the MIB ifName. Each ifName varies according to the vendor's conventions. Using the name `4/1` as an example, `4` represents `the board number and 1` represents the port number.
$sourceObjectUuid, $sou	Universally Unique Object Identifier attribute value of the source object for the incident (unique across all databases). This identifier

Valid Parameters Not Visible From a Form, continued

Parameter Value	Description
	distinguishes the source object instance from all other similar object instances..
$uuid	Universally Unique Object Identifier attribute value of the incident (unique across all databases). This identifier distinguishes the incident object instance from all other incident objects.

Valid Parameters Established in Custom Incident Attributes

Parameter Value	Description
$<position_ number>	Value of the custom incident attribute (CIA) position number for any CIA that originated from a varbind or was added by NNMi. For example, to indicate you want to use the varbind in position 1, enter: `$1` NNMi stores varbind values as Custom Incident Attributes. If you know the varbind position number, use this parameter.
$<CIA_ name>	Value of the name that is used for the custom incident attribute. For example, `$mycompany.mycia`. NNMi provides CIA values for configuring Management Events. See Custom Incident Attributes Provided by NNMi for more information about custom incident attributes.
$<CIA_ oid>	Value of the object identifier for any custom incident attribute that originated as a varbind. For example, `$.1.3.6.1.6.3.1.1.5.1`. Use this parameter when you are not certain of a custom incident attribute (varbind) position number.
$*	Used to indicate you want all of the custom incident attribute values originating as varbinds, to be passed to the action configuration. Each varbind is returned in the following format: `$<`*`CIA_name`*`>:<`*`CIA_value`*`>` in which the custom incident attribute name appears followed by the custom incident attribute value.

The function described in the following table replaces the specified numeric value with the associated text value stored in the CIA.

Note: The associated MIB must have been loaded using the nnmloadmib.ovpl command.

Functions to Generate Values Within Incident Messages

Function	Description
$text ($<position_ number>)	The <*position_number*> argument specifies the numeric value of the custom incident attribute (CIA) position number for any CIA that originated from a varbind or was added by NNMi. For example, to indicate you want to use the varbind in position 1, enter: `$1`. After the function runs, NNMi replaces the numeric value with the text value

Functions to Generate Values Within Incident Messages, continued

Function	Description
	stored in the CIA. **Note**: If a text value is not available, NNMi returns the numeric value.
$text ($<CIA_ oid>)	The <CIA_oid> argument specifies the object identifier for any custom incident attribute that originated as a varbind. For example, `$.1.3.6.1.6.3.1.1.5.1.` Use this argument to the $text function when you are not certain of a custom incident attribute (varbind) position number. After the function runs, NNMi replaces the numeric value with the text value stored in the CIA. **Note**: If a text value is not available, NNMi returns the numeric value.

Configure Remote NNM 6.x/7.x Events

NNMi can display incidents from Remote NNM 6.x and 7.x management stations. In the NNMi incident browser, you can manage the lifecycle of incidents generated from previous versions of NNMi.

Tip: Gradually upgrade from NNM 6.x or 7.x to NNMi while using this feature.

To configure NNMi to handle incidents generated from remote NNM 6.x/7.x events, perform the following tasks:

- Configure the NNM 6.x/7.x Management Stations
- Configure what NNMi does with the NNM 6.x/7.x events

Configure Remote NNM 6.x and 7.x Management Stations

There are multiple benefits to configuring NNMi to recognize the NNM 6.x or 7.x management stations in your environment:

- Configure NNMi to receive and display incidents (events) from remote NNM 6.x or 7.x management stations.
- Enable displaying NNM 6.x or 7.x Dynamic Views from forwarded NNM 6.x or 7.x events (see Access NNM 6.x and 7.x Features for more information).
- Filter NNMi view by NNM 6.x or 7.x management station (show only those incidents received from a particular NNM 6.x or 7.x management station).

To display the details of an NNM 6.x or 7.x management station configuration:

1. From the workspace navigation panel, select the **Inventory** workspace.
2. Select the **Management Stations (6.x/7.x)** view.
3. Double-click the row representing the configuration you want to edit.

The Management Station form displays.

4. When finished, click the Close icon.

To configure an NNM 6.x or 7.x management station:

Note: Your User Account must be assigned to the **NNMi Administrators** User Group to perform this task.

NNMi Advanced. Your NNMi management server must be configured as either an IPv4 or dual stack (IPv4/IPv6) machine to proceed. You must configure an IPv4 address for communication between an NNMi management server and the remote NNM 6.x or 7.x management station. (See the NNMi Advanced Release Notes for details.)

1. Navigate to the **Management Stations (6.x/7.x)** view.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Select the **Management Stations (6.x/7.x)** view.
2. Do one of the following:
   - To create an NNM 6.x or 7.x management station configuration, click the * New icon, and continue.
   - To edit an NNM 6.x or 7.x management station configuration, double-click the row representing the configuration you want to edit, and continue.
   - To delete an NNM 6.x or 7.x management station configuration, select a row, and click the ✗ Delete icon.
3. In the Management Station form, provide the required information:
   - IPv4 address of the remote NNM 6.x or 7.x management station
   - Port number used by the OpenView Application Server (ovas) on the remote NNM 6x or 7x management station
   - Port number used by the web server on the remote NNM 6x or 7x management station
4. Click **Save and Close** to return to the Management Stations (6.x/7.x) view.
5. If this is the first Management Station configuration, you must exit the NNMi console, and start the NNMi console. (You do not need to exit and start the NNMi console when configuring any subsequent NNM 6.x/7.x management stations.)
6. Next, configure which incidents to receive from your NNM 6.x or 7.x management station ("Configure Remote NNM 6.x/7.x Events" on the previous page).

Remote NNM 6.x/7.x Event Configuration Form

Using NNMi, you can display incidents from Remote NNM 6.x and 7.x management stations . In the NNMi incident browser, you can manage the lifecycle of incidents generated from previous versions of NNMi.

Tip: Gradually upgrade from NNM 6.x or 7.x to NNMi while using this feature.

To configure a Remote NNM 6.x/7.x event:

1. Navigate to the **Remote NNM 6.x/7.x Event Configuration** form:
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Click to expand the **Incidents** folder.
   c. Select **Remote NNM 6.x/7.x Event Configurations** .
   d. Do one of the following:
      - To create a Remote NNM 6.x/7.x Event configuration, click the ✳ New icon.
      - To edit a Remote NNM 6.x/7.x Event configuration, select a row, click the 📂 Open icon, and continue.

   Note: In the Remote NNM 6.x/7.x Event Configuration form, verify that **Enable**☑ is selected.
2. Make your configuration choices (see table).
3. Click 💾 **Save and Close** to save your changes and return to the **Incident Configuration** form.

Tasks for Remote NNM 6.x/7.x Event Configuration

Task	How
"Specify the Incident Configuration Name (Remote 6.x/7.x Event)" on page 1227	Use the **Basics** group of the Remote NNM 6.x/7.x Event Configuration form. Specify a name that helps you to identify the configuration for subsequent use.
Specify whether you want to enable this configuration.	In the **Basics** group of the Remote NNM 6.x/7.x Event Configuration form, make sure **Enable**☑ is checked for each configuration you want to use.
Display the NNMi Remote Incident as a Root Cause Incident	Use the **Basics** group of the Remote NNM 6.x/7.x Event Configuration form.
"Specify Category and Family Attribute Values for Organizing Your Incidents (Remote NNM 6.x/7x Events)" on page 1228	Use the **Basics** group of the Remote NNM 6.x/7.x Event Configuration form. You can organize your incidents using Category and Family.
"Specify the Incident Severity (Remote NNM 6.x/7.x Events)" on page 1232	Use the **Basics** group of the Remote NNM 6.x/7.x Event Configuration form. Possible Severity values include: **Normal, Warning, Minor, Major,** and **Critical**.
"Specify Your Incident Message Format (Remote NNM 6.x/7.x Events)" on page 1233	Use the **Basics** group of the Remote NNM 6.x/7.x Event Configuration form. The message format determines the message to be displayed for the incident.
"Specify a Description for Your Incident Configuration (Remote NNM 6.x/7.x Events)" on page 1240	Use the **Basics** group of the Remote NNM 6.x/7.x Event Configuration form. Provide a meaningful description.

Tasks for Remote NNM 6.x/7.x Event Configuration, continued

Task	How
Specify an Author for Your Remote NNM 6.x/7.x Event Configuration	Use the **Basics** pane of the **Remote NNM 6.x/7.x Event Configuration** form to indicate who created or last modified the event. **Caution:** If the Author attribute value is **HP Network Node Manager**, any changes are at risk of being overwritten in the future. • Click **Lookup** and select **Show Analysis** to display details about the currently selected Author. • Click **Quick Find** to access the list of existing Author values. • Click ✱ **New** to create an Author value.

After you complete the Basic Configuration for the remote NNM 6.x or 7.x event, you can also choose to configure the information described in the following table.

Additional Configurations

Task	How
"Correlate Duplicate Incidents (Deduplication Configuration)" on page 659	Select the **Deduplication** tab to specify duplicate incidents that you want to be suppressed.
"Track Incident Frequency (Rate: Time Period and Count)" on page 659	Select the **Rate** tab to specify a rate for duplicate incidents. After the rate limit is reached, NNMi generates an Incident to notify you of the problem.
"Configure an Action for an Incident" on page 748	Select the **Actions** tab to specify actions that should occur automatically when an incident changes its Lifecycle State.
"Configure Diagnostics for an Incident (NNM iSPI NET)" on page 757	Select the **Node Settings** tab to specify diagnostic actions that should occur automatically when an incident reaches a selected Lifecycle State for a node that belongs to a particular Node Group.

Configure Basic Settings for a Remote NNM 6.x/7.x Event Incident

The Basics settings for a Remote NNM 6.x/7.x event incident specifies general information for an incident configuration, including the name, severity, and message.

Note: In the **Basics** group of the **Remote NNM 6.x/7.x Event Configuration** form, verify that **Enable** ☑ is selected for each configuration you want to use.

For information about each Remote NNM 6.x/7.x Events tab:

To configure Basic settings for a Remote NNM 6.x/7.x Event incident:

Navigate to the **Remote NNM 6.x/7.x Event Configuration** form:

1. From the workspace navigation panel, select the **Configuration** workspace.
2. Expand the **Incidents** folder.
3. Select **Remote NNM 6.x/7.x Event Configurations**.
4. Do one of the following:
   a. To create an incident configuration, click the ✱ New icon, and continue.
   b. To edit an incident configuration, select a row, click the Open icon, and continue.
   c. To delete an incident configuration, select a row and click the ✖ Delete icon.
5. Configure the required Basic settings (see table).
6. Click **Save and Close** to save your changes and return to the previous form.

Basics Attributes for SNMP Trap Configuration

Task	How
"Specify the Incident Configuration Name (Remote 6.x/7.x Event)" on page 1227	Use the **Basics** pane of the **Remote NNM 6.x/7.x Event Configuration** form. Specify a name that helps you to identify the configuration for subsequent use.
Specify whether you want to enable this configuration.	In the **Basics** group of the **Remote NNM 6.x/7.x Event Configuration** form, verify that **Enable**☑ is selected for each configuration you want to use.
"Display an SNMP Trap or an NNM 6.x/7.x Event as a Root Cause Incident" on page 789	Use the **Basics** pane of the **Remote NNM 6.x/7.x Event Configuration** form.
"Specify Category and Family Attribute Values for Organizing Your Incidents (Remote NNM 6.x/7x Events)" on page 1228	Use the **Basics** pane of the **Remote NNM 6.x/7.x Event Configuration** form. You can organize your incidents using Category and Family.
"Specify the Incident Severity (Remote NNM 6.x/7.x Events)" on page 1232	Use the **Basics** pane of the **Remote NNM 6.x/7.x Event Configuration** form. Possible Severity values include: **Normal, Warning, Minor, Major,** and **Critical**.
"Specify Your Incident Message Format (Remote NNM 6.x/7.x Events)" on page 1233	Use the **Basics** pane of the **Remote NNM 6.x/7.x Event Configuration** form. The message format determines the message to be displayed for the incident.
"Specify a Description for Your Incident Configuration (Remote NNM 6.x/7.x	Use the **Basics** pane of the **Remote NNM 6.x/7.x Event Configuration** form. Provide a meaningful

Basics Attributes for SNMP Trap Configuration, continued

Task	How
Events)" on page 1240	description.
Specify an Author for Your Incident Configuration (Remote NNM 6.x/7.x Events)	Use the **Basics** pane of the **Remote NNM 6.x/7.x Event Configuration** form to indicate who created or last modified the event. **Caution:** If the Author attribute value is **HP Network Node Manager**, any changes are at risk of being overwritten in the future. • Click **Lookup** and select **Show Analysis** to display details about the currently selected Author. • Click **Quick Find** to access the list of existing Author values. • Click **New** to create an Author value.

After you complete the Basic Configuration for the remote NNM 6.x/7.x event, you can also choose to configure the information described in the following table.

Additional Incident Configurations

Task	How
"Configure Interface Settings for a Remote NNM 6.x/7.x Event Incident" on page 1241	Select the **Interface Settings** tab to specify an Interface Group to which you want your incident configuration to apply.
"Configure Node Settings for a Remote NNM 6.x/7.x Event Incident" on page 1278	Select the **Node Settings** tab to specify a Node Group to which you want your incident configuration to apply.
"Configure Suppression Settings for a Remote NNM 6.x/7.x Event Incident" on page 1317	Select the **Suppression** tab to specify the criteria for discarding incidents that match the selected incident configuration.
"Configure Enrichment Settings for a Remote NNM 6.x/7.x Event Incident" on page 1325	Select the **Enrichment** tab to specify enhancements for the selected incident configuration.
"Configure Dampening Settings for a Remote NNM 6.x/7.x Event Incident" on page 1329	Select the **Dampen** tab to specify the time interval that must be met before the incident appears in an Incident view.
"Configure Deduplication for a Remote NNM 6.x/7.x Event Incident" on page 1346	Select the **Deduplication** tab to specify duplicate incidents that you want to be suppressed.

Additional Incident Configurations, continued

Task	How
"Configure Rate (Time Period and Count) for a Remote NNM 6.x/7.x Event Incident" on page 1353	Select the **Rate** tab to specify a rate for duplicate incidents. After the rate limit is reached, NNMi generates an Incident to notify you of the problem.
"Configure Actions for a Remote NNM 6.x/7.x Event Incident" on page 1357	Select the **Actions** tab to specify actions that should occur automatically when an incident changes its Lifecycle State.

Specify the Incident Configuration Name (Remote 6.x/7.x Event)

When providing the Name for an incident configuration, use the following guidelines:

Name

The name is used to identify the incident configuration and must be unique. Use a name that will help you to remember the purpose or kind of management event, remote NNM 6.x or 7.x event or SNMP trap, for which you are configuring an incident. Name is also used to identify your Pairwise configurations.

Alpha-numeric and special characters (~ ! @ # $ % ^ & * ( ) _+ -) are permitted. No spaces are permitted.

Display an SNMP Trap or an NNM 6.x/7.x Event as a Root Cause Incident

SNMP trap and NNM 6.x/7.x events normally appear as symptoms rather than as root cause incidents. However, there might be times when you want an SNMP or NNM 6.x/7.x event to appear as a root cause incident. For example, you might want an HSRP state change (cHsrpStateChange, 1.3.6.1.4.1.9.9.106.2.0.1) trap to be listed as a root cause. This trap might occur when the hot standby has gone down indicating the system is at risk if there is a failover.

Note: To reduce "noise" associated with duplicate incidents, NNMi changes the incident Correlation Nature to **Symptom** for any user-defined Root Cause incidents that exceed the rate or deduplication threshold.

To display an SNMP trap or NNM 6.x/7.x Event as a root cause incident:

Select **Root Cause** ☑ in the **SNMP Trap** or **Remote NNM 6.x/7.x Event Configuration** form.

The next time the incident occurs, the Correlation Nature attribute value changes to **Root Cause**.

To no longer display an SNMP trap or NNM 6.x/7.x Event s a root cause incident:

Clear **Root Cause** □ in the **SNMP Trap** or **Remote NNM 6.x/7.x Event Configuration** form.

The next time the incident occurs, the Correlation Nature attribute value changes to **Symptom**.

Specify Category and Family Attribute Values for Organizing Your Incidents (Remote NNM 6.x/7x Events)

When configuring incidents, NNMi provides the Category and Family attributes to help you organize your incidents.

Preconfigured Categories
The Category attribute helps you organize your incidents. Select the category that you want to be associated with this type of incident when it appears in an incident view. Each of the possible Category values is described in the following table.

Incident Categories Provided by NNMi

Category	Description
Accounting	Used to indicate problems related to usage statistics and allocation of costs associated with the billing of time and services provided by devices. This category is not used by NNMi with default configurations, but it is available for incidents you define.
Application Status	Indicates there is a problem with the health of the NNMi software. Examples of these kinds of events include license expiration (see "Extend a Licensed Capacity" on page 1575) or that a certain NNMi process or service lost connection to the Process Status Manager (see "Stop or Start an NNMi Process" on page 82 and "Stop or Start NNMi Services" on page 86).
Configuration	Indicates there is a problem with the configuration of a managed device. For example, there is a physical address mismatch.
Fault	Indicates a problem with the network, for example Node Down.
Performance	Indicates a Monitored Attribute value *crossed* a configured threshold. For example, Disk Space Utilization exceeds the configured threshold criteria for High Value = 90 percent .
Security	Indicates there is a problem related to authentication. For example, an SNMP authentication failure.
Status	Indicates some kind of status message. Examples of these kinds of incidents include "SNMP Link Up" or an "HSRP Group status Normal" message.

Note: You can add your own Category entries to NNMi. See "Create an Incident Category (Remote NNM 6.x/7.x Event)" on page 1230 for more information.

You can use Family values to further categorize the types of incidents that might be generated. Each of the possible Family values are described in the following table.

Incident Family Attribute Values Provided by NNMi

Family	Description
Address	Indicates the incident is related to an address problem.

Incident Family Attribute Values Provided by NNMi, continued

Family	Description
Aggregated Port	Indicates the incident is related to a **Link Aggregation**[1] problem.
BGP	Indicates the incident is related to a problem with BGP (Border Gateway Protocol). This family is not used by NNMi with default configurations, but it is available for incidents you define.
Board	Indicates the incident is related to a board problem. This family is not used by NNMi with default configurations, but it is available for incidents you define.
Chassis	Indicates the incident is related to an board problem. This family is not used by NNMi with default configurations, but it is available for incidents you define.
Component Health	Indicates the incident is related to Node Component metrics collected by NNMi. See Node Form: Node Component Tab for more information about the Node Component metrics collected.
Connection	Indicates the incident is related to a problem with one or more connections.
Correlation	Indicates the incident has additional incidents correlated beneath it. These incidents are associated with a duplicate count so that you can determine the number of correlated incidents associated with it.
Custom Poller	Indicates the incident is related to the NNMi Custom Poller feature. See About Custom Poller.
HSRP	*NNMi Advanced*. Indicates the incident is related to a problem with Hot Standby Router Protocol (**HSRP**[2]).
Interface	Indicates the incident is related to a problem with one or more interfaces.
License	Indicates the incident is related to a licensing problem.
NNMi Health	Indicates the incident is related to NNMi Health. See the Check NNMi Health for more information.
Node	Indicates the incident is related to a node problem.
OSPF	Indicates the incident is related to an OSPF problem. This family is not used by NNMi with default configurations, but it is available for incidents you define.
RAMS	*NNMi Advanced*. Indicates the incident is related to a Router Analytics

[1] Protocols used on Switches to configure multiple Interfaces (Aggregation Member Interfaces) to function as if they were one (an Aggregator Interface). When two Aggregator Interfaces establish a connection, that connection is an Aggregator Layer 2 Connection. The Aggregator Layer 2 Connection appears on Layer 2 Neighbor View maps as a thick line with an Interface icon at each end (representing the Aggregator Interface). When you double-click the thick line, it converts into multiple thin lines representing the participating Aggregation Member Layer 2 Connections with their Aggregation Member Interfaces at each end of the lines.

[2] Hot Standby Router Protocol

Incident Family Attribute Values Provided by NNMi, continued

Family	Description
	Management System problem.
RMON	Indicates the incident is related to a Remote Monitor (IETF standard, RFC 1757) problem. This family is not used by NNMi with default configurations, but it is available for incidents you define.
RRP	*NNMi Advanced*. Indicates the incident is related to a problem with a Router Redundancy Protocol configuration.
STP	Indicates the incident is related to Spanning-Tree Protocol problem. This family is not used by NNMi with default configurations, but it is available for incidents you define.
Syslog	NNMi does not use this Family with default configurations. It is available for incidents you define.
Trap Analysis	Indicates the incident is related to an SNMP trap storm.
VLAN	Indicates the incident is related to a problem with a virtual local area network.
VRRP	*NNMi Advanced*. Indicates the incident is related to a problem with Virtual Router Redundancy Protocol (**VRRP**[1]).

Note: You can add your own Family entries to NNMi. See "Create an Incident Family (Remote NNM 6x./7.x Event)" on the next page for more information.

Create an Incident Category (Remote NNM 6.x/7.x Event)

The Category attribute helps you organize your incidents. Create any Category that makes sense to you and your team. For a list of the Category codes provided by NNMi, "Specify Category and Family Attribute Values for Organizing Your Incidents (Remote NNM 6.x/7x Events)" on page 1228.

To create a new incident Category:

1. Navigate to the **Incident Category** form.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Expand the **Incidents** folder.
   c. Select **Remote NNM 6.x/7.x Event Configurations**.
   d. Do one of the following:
      - To create an incident configuration, click the New icon.
      - To edit an incident configuration, select a row, click the Open icon, and continue.
   e. In the configuration form, locate the **Category** attribute.

[1] Virtual Router Redundancy Protocol

f. Click the Lookup icon, and select ✱ New.

2. Provide the required information (see table).

3. Click **Save and Close** to save your changes and return to the previous form.

Category Code Attributes

Name	Description
Label	Incident category name. For example, Hardware Faults, or Cisco Error. Maximum size is 255 characters. Alpha-numeric, spaces, and underline characters are permitted.

Unique Key	**Caution**: After you click **Save and Close**, this value cannot be changed.
	Used as a unique identifier when exporting and importing configuration definitions. To ensure that the value you enter is unique, it is recommended that you use the Java name space convention when providing this value. It is also useful to include the label value as part of the unique key as shown in the following examples:
	`com.<your_company_name>.nnm.trapConf.category.<category_label>`
	`com.<your_company_name>.nnm.eventConf.category.<category_label>`
	`com.<your_company_name>.nnm.inciConf.category.<category_label>`
	The maximum length is 80 characters. Alpha-numeric characters and periods are permitted. Spaces are not permitted.

Create an Incident Family (Remote NNM 6x./7.x Event)

The Family attribute helps you organize your incidents. Create any Family that makes sense to you and your team. For a list of the Family codes provided by NNMi, "Specify Category and Family Attribute Values for Organizing Your Incidents (Remote NNM 6.x/7x Events)" on page 1228.

To create a new incident Family:

1. Navigate to the **Incident Family** form.

   a. From the workspace navigation panel, select the **Configuration** workspace.

   b. Expand the **Incidents** folder.

   c. Select **Remote NNM 6.x/7.x Event Configurations**.

   a. Do one of the following:

      - To create an incident configuration, click the ✱ New icon.

      - To edit an incident configuration, select a row, click the Open icon, and continue.

   b. In the configuration form, locate the **Family** attribute.

   c. Click the Lookup icon, and select ✱ New.

2. Provide the required information (see table).
3. Click **Save and Close** to save your changes and return to the previous form.

Family Attributes

Name	Description
Label	Family name. For example, Hardware Faults, or Cisco Error. Maximum size is 255 characters. Any character type is valid.

Unique Key	**Caution**: After you click **Save and Close**, this value cannot be changed.
	Used as a unique identifier when exporting and importing configuration definitions. To ensure that the value you enter is unique, it is recommended that you use the Java name space convention when providing this value. It is also useful to include the label value as part of the unique key as shown in the following examples:
	`com.<your_company_name>.nnm.trapConf.family.<family_label>`
	`com.<your_company_name>.nnm.eventConf.family.<family_label>`
	`com.<your_company_name>.nnm.inciConf.family.<family_label>`
	The maximum length is 80 alpha-numeric characters, periods allowed, no spaces allowed.

Specify the Incident Severity (Remote NNM 6.x/7.x Events)

The incident severity represents the seriousness calculated for the incident. Use the severity attribute to specify the severity that should be assigned to the incident you are configuring. Possible values are described in the following table.

Incident Severity Values

Attribute	Description
Normal	Indicates there are no known problems related to the associated object. This severity is meant to be informational. Generally, no action is needed for these incidents.
Warning	Indicates there might be a problem related to the associated object.
Minor	Indicates NNMi has detected problems related to the associated object that require further investigation.
Major	Indicates NNMi has detected problems related to the associated object to be resolved before they become critical.
Critical	Indicates NNMi has detected problems related to the associated object that require immediate attention.

See "Monitor Incidents for Problems" for more information about these severity values.

Specify Your Incident Message Format (Remote NNM 6.x/7.x Events)

When configuring an incident, specify the information you want NNMi to include in the incident's Message attribute value. You can use any combination of valid parameter strings and Custom Incident attributes to configure the Message.

Note: The incident Message limit is 1024 characters. If the returned values exceed this limit, NNMi truncates the value starting from the end of the returned text string.

"Valid Parameters for Configuring Incident Messages (Remote NNM 6.x/7.x Events)" below

"Include Custom Incident Attributes in Your Message Format (Remote NNM 6.x/7.x Events)" on page 1239

Valid Parameters for Configuring Incident Messages (Remote NNM 6.x/7.x Events)

When configuring incident messages, consider using incident information as part of the message. NNMi provides the following parameter values. Use these parameters as variables when formatting an incident message.

Tip: See the Using the Incident Form for more information about the parameter values.

Note: NNMi stores varbind values as custom incident attributes (CIAs).

Tip: If a value is not stored for a parameter, it is returned as “null”.

See "Specify Your Incident Message Format (Remote NNM 6.x/7.x Events)" above for more information about configuring messages.

Parameter strings are available for the following:

Note: See the following tables to view the valid parameters for incidents generated from Custom Polled Instances: Parameter Strings for all Incidents (Attributes from an Incident form), Parameter Strings for Node Source Objects (Attributes from a Node form), and the Parameter Strings for all Incidents (Attributes not Visible from any form).

- Parameter strings for all incidents (Incident form attributes) (Click here for a list of choices.)

Parameter Strings for all Incidents (Incident form attributes)

Parameter String	Description
$category, $cat	Value of the Category attribute in the Incident form.
$count, $cnt	Value representing the number of Custom Incident Attributes that appear in the Incident form.
$family, $fam	Value from the Family attribute in the Incident form.
$firstOccurrenceTime, $fot	Value from the First Occurrence Time attribute in the incident form.
$lastOccurrenceTime, $lot	Value from the Last Occurrence Time attribute in the incident form.
$lifecycleState, $lcs	Value from the Lifecycle State attribute in the Incident form.
$name	Value of the Name attribute from the incident configuration.
$nature, $nat	Value from the Nature attribute in the Incident form.
$origin, $ori	Value from the Origin attribute in the Incident form.
$originOccurrenceTime, $oot	Value from the Origin Occurrence Time attribute in the incident form.
$priority, $pri	Value from the Priority attribute in the Incident form.
$sev, $severity	Value of the Severity attribute of the Incident form.

- Parameter Strings for Node Source Objects (Node form attributes) (Click here for a list of choices.)

Parameter Strings for Node Source Objects (Node form attributes)

Parameter String	Description
$managementAddress, $mga	Value from the Management Address attribute of the incident's source Node's form or SNMP Agent form.
$otherSideOfConnectionManagementAddress, $oma	If the incident's Source Node is part of a Layer 2 Connection, this attribute is the value of the Management Address of a node on the other side of the Layer 2 Connection.
$sourceNodeLongName, $sln	The fully-qualified DNS name as displayed in the Hostname attribute of the incident's source Node's form.
$sourceNodeName, $snn	Value from the Name attribute of the incident's source Node's form.

Parameter Strings for Node Source Objects (Node form attributes) , continued

Parameter String	Description
$sysContact, $sct	Value from the System Contact attribute of the incident's source Node form: General tab.
$sysLocation, $slc	Value from the System Location attribute of the incident's source Node form: General tab.

- Parameter Strings for Interface Source Objects (Interface form attributes) (Click here for a list of choices.)

Parameter Strings for Interface Source Objects (Interface form attributes)

Parameter String	Description
$ifAlias, $ifa	Value from the IfAlias attribute for the interface that is the incident's source object.
$ifConfigDupSetting, $icd	Configured Duplex Setting on the port associated with the interface that is the incident's source object.
$ifDesc, $idc	Value from the ifDesc attribute for the interface that is the incident's source object.
$ifIndex, $idx	Value from the ifIndex attribute for the interface that is the incident's source object.
$ifIpAddr, $iia	IP Address values associated with the interface that is the incident's source object. If multiple IPaddresses are associated with the interface, this parameter returns a comma-separated list.
$ifName, $ifn	Value from the ifName attribute for the interface that is the incident's source object.
$ifPhysAddr, $ipa	Value from the Physical Address attribute for the interface that is the incident's source object.
$ifSpeed, $isp	Value from the ifSpeed attribute for the interface that is the incident's souce object.
$ifType, $itp	Value from the ifType attribute for the interface that is the incident's souce object.

- Parameter Strings for Layer 2 Connection Source Objects (Layer 2 Connection form attributes) (Click here for a list of choices.)

Parameter Strings for Layer 2 Connection Source Objects (Layer 2 Connection form attributes)

Parameter String	Description
$otherSideOfConnectionConfigDupSetting, $ocd	If the incident's source Node is part of a Layer 2 Connection, this parameter contains the Configured Duplex Setting on the port associated with the interface on the other side of the connection.
$otherSideOfConnectionIfAlias, $oia	If the incident's Source Node is part of a Layer 2 Connection, this parameter is the value of the ifAlias of one of the interfaces on the other side of the Layer 2 Connection.
$otherSideOfConnectionIfDesc, $odc	If the incident's Source Node is part of a Layer 2 Connection, this parameter contains the ifDescr attribute value for the interface on the other side of the Layer 2 Connection.
$otherSideOfConnectionIfIndex, $odx	If the incident's Source Node is part of a Layer 2 Connection, this parameter contains the ifIndex attribute value for the interface on the other side of the connection.
$otherSideOfConnectionIfName, $ofn	If the incident's Source Node is part of a Layer 2 Connection, this parameter contains the ifName attribute value for the interface on the other side of the connection.

- Parameter strings for VLAN Source Objects (VLAN form attributes) (Click here for a list of choices.)

Parameter Strings for VLAN Source Objects (VLAN form attributes)

Parameter String	Description
$impVlanIds, $ivi	Value from the VLAN Id attribute associated with the interface that is the incident's source object. To access this information from an interface form, navigate to the VLAN Port tab and open the form for the VLAN of interest. If the interface is part of more than one VLAN, this parameter returns a comma-separated list.
$impVlanNames, $ivn	Value from the VLAN Name attribute associated with the interface that is the incident's source object. To access this information from an interface form, navigate to the VLAN Ports tab of the Interface form. If the interface is part of more than one VLAN, this parameter returns a comma-separated list.

- Parameter Strings for all incidents (Additional information that is not visible in any form) (Click here for a list of choices.)

Parameter Strings for all Incidents (Attributes not visible in any form)

Parameter String	Description
$firstOccurrenceTimeMs, $fms	Value from the First Occurrence Time attribute in the incident form, converted to millseconds (measured since January 1, 1970, 00:00:00 GMT - Greenwich Mean Time).
$lastOccurrenceTimeMs, $lms	Value from the Last Occurrence Time attribute in the incident form, converted to millseconds (measured since January 1, 1970, 00:00:00 GMT - Greenwich Mean Time).
$oid	Value of the unique object identifier (oid) for the incident configuration that originated from either an SNMP trap, Management Event, or Remote NNM 6.x or 7.x event.
$otherSideOfConnection, $osc	If the incident's Source Node is part of a Layer 2 Connection, this attribute is the following combination of values for the node and one of its interfaces on the other side of the Layer 2 Connection: The fully-qualified DNS name of the node appended with the interface Name in the following format: <*fully-qualified DNS name*>[*interface_name*]
$originOccurrenceTimeMs $oms	Value from the Origin Occurrence Time attribute in the incident form, converted to millseconds (measured since January 1, 1970, 00:00:00 GMT - Greenwich Mean Time).
$sourceNodeUuid, $snu	Universally Unique Object Identifier attribute value of the source node object for the incident (unique across all databases). This identifier distinguishes the source node object instance from all other node objects.
$sourceObjectClass, $soc	Value of the object class for the object you want to include. Use this parameter to request more details of a class of objects through a web service. Examples of object classes include: `com.hp.ov.nms.model.core.Interface` and `com.hp.ov.nms.model.snmp.SnmpAgent.`
$sourceObjectName, $son	Value from the Name attribute of the source object. For example, an interface object is named according to the MIB ifName. Each ifName varies according to the vendor's conventions. Using the name `4/1` as an example, `4` represents `the board number and 1` represents the port number.
$sourceObjectUuid, $sou	Universally Unique Object Identifier attribute value of the source object for the incident (unique across all databases). This identifier distinguishes the source object instance from all other similar object instances.
$uuid	Universally Unique Object Identifier attribute value of the

Parameter Strings for all Incidents (Attributes not visible in any form), continued

Parameter String	Description
	incident (unique across all databases). This identifier distinguishes the incident object instance from all other incident objects.

- Information established in Custom Incident Attributes (Click here for a list of choices.)

Parameter Strings for Attributes Established in Custom Incident Attributes

Parameter String	Description
$<position_number>	Value of the custom incident attribute (CIA) position number for any CIA that originated from a varbind or was added by NNMi. For example, to indicate you want to use the varbind in position 1, enter: `$1` NNMi stores varbind values as Custom Incident Attributes. If you know the varbind position number, use this parameter.
$<CIA_name>	Value of the name that is used for the custom incident attribute. For example, `$mycompany.mycia`. NNMi provides CIA values for configuring Management Events. See Custom Incident Attributes Provided by NNMi for more information about custom incident attributes.
$<CIA_oid>	Value of the object identifier for any custom incident attribute that originated as a varbind. For example, `$.1.3.6.1.6.3.1.1.5.1`. Use this parameter when you are not certain of a custom incident attribute (varbind) position number.
$*	Used to indicate you want all of the custom incident attribute values originating as varbinds, to be passed to the action configuration. Each varbind is returned in the following format: `$<CIA_name>:<CIA_value>` in which the custom incident attribute name appears followed by the custom incident attribute value.

- Functions to generate values (Click here for a list of choices.)

The function described in the following table replaces the specified numeric value with the associated text value stored in the CIA.

Note: The associated MIB must have been loaded using the nnmloadmib.ovpl command.

Functions to Generate Values Within the Incident Message

Function	Description
$oidtext($<position_number>)	A <*position_number*> argument specifies the numeric value of the custom incident attribute (CIA) position number for any CIA that originated from a varbind or was added by NNMi. For example, `$oidtext($2)`. **Note**: The position number you enter must represent a CIA that contains an Object Identifier (OID) value. NNMi returns the textual value of the OID for the CIA specified.

Functions to Generate Values Within the Incident Message, continued

Function	Description
	Note the following: ▪ If the MIB is not loaded, NNMi returns the numeric OID value. ▪ If the OID has a MIB instance, the number representing the MIB instance is appended to the textual OID value.
$oidtext ($<CIA_ oid>)	The <*CIA_oid*> argument specifies the Object Identifier (OID) for any custom incident attribute that originated as a varbind. For example, `$oidtext ($.1.3.6.1.6.3.1.1.5.1.)` Use this argument to the $oidtext() function when you are not certain of a custom incident attribute (varbind) position number. NNMi replaces the numeric value with the textual value of the OID you specify. Note the following: ▪ If the MIB is not loaded, NNMi returns the numeric OID value. ▪ If the OID has a MIB instance, the number representing the MIB instance is appended to the textual OID value.
$text ($<position_ number>)	The <*position_number*> argument specifies the numeric value of the custom incident attribute (CIA) position number for any CIA that originated from a varbind or was added by NNMi. For example, to indicate you want to use the varbind in position 1, enter: `$1.` NNMi replaces the numeric value with the text value stored in the CIA. **Note**: If a text value is not available, NNMi returns the numeric value.
$text ($<CIA_ oid>)	The <CIA_oid> argument specifies the object identifier for any custom incident attribute that originated as a varbind. For example, `$.1.3.6.1.6.3.1.1.5.1.` Use this argument to the $text function when you are not certain of a custom incident attribute (varbind) position number. NNMi replaces the numeric value with the text value stored in the CIA. **Note**: If a text value is not available, NNMi returns the numeric value.

Include Custom Incident Attributes in Your Message Format (Remote NNM 6.x/7.x Events)

NNMi includes two categories of CIAs:

- SNMP trap varbinds identified by the Abstract Syntax Notation value (ASN.1). NNMi turns varbinds into CIAs and maintains each varbind's position number. See "Load SNMP Trap Incident Configurations" on page 771.
- Custom incident attributes provided by NNMi. See "Custom Incident Attributes Provided by

NNMi (Information for Administrators)" on page 647.

You cannot create Custom Incident Attributes.

You can use CIAs in your message format to extend the amount of information presented. To determine which CIAs are available for any particular incident type, open an Incident view, locate the incident and open the Incident form. Navigate to the **Custom Attributes** tab. A complete list of available CIAs (for that incident type) appears in the table.

To include a CIA in your message format, type the dollar-sign character (`$`) plus any of the following:

- Varbind position number or asterisk (*) to include all varbind values
- Name of the CIA
- Object identifier (oid) of the CIA (useful when the varbind position number is not consistent among vendors)

Note: A single incident cannot include two CIAs with the same name. However, two incidents can contain CIAs having the same names and values.

The following table presents some example formats with the subsequent output.

Example Incident Message Formats

Example Message Format	Output in Incident View
Possible trouble with $3	`Possible trouble with` <varbind 3>
Possible trouble with $11	`Possible trouble with` <varbind 11>
Possible trouble with $77 (where the varbind position 77 does not exist)	`Possible trouble with <Invalid or unknown cia> 77`
Possible trouble with $*	`Possible trouble with` <cia1_name: cia_value>, <cia2_name; cia_value>,< cia*n*_name: cia_value>
Possible trouble with $3x	`Possible trouble with` <varbind 3>`x`
Possible trouble with $1.2.3.4.5	`Possible trouble with` <value of the CIA with oid of 1.2.3.4.5>
Possible trouble with $mycia.mycompany	`Possible trouble with` <value of the CIA with name of mycia.mycompany>

Tip: NNMi provides an error message when a CIA cannot be found. For example, if you enter an unavailable varbind position, name, or object identifier (oid), NNMi returns an "Invalid or unknown cia" error message.

Specify a Description for Your Incident Configuration (Remote NNM 6.x/7.x Events)

NNMi provides the Description attribute to help you further identify the current incident configuration.

Description

Use the description field to provide additional information that you would like to store about the current incident configuration. This description applies only to the configuration entry.

Type a maximum of 1024 characters. Alpha-numeric, spaces, and special characters (~ ! @ # $ % ^ & * ( ) _+ -) are permitted.

Configure Interface Settings for a Remote NNM 6.x/7.x Event Incident

Note: Interface Settings override any other Suppression, Enrichment, Dampen, or Actions settings for this incident, including those configured on the Node Settings tab.

NNMi enables you to apply an incident configuration to a Source Object based on the Source Object's participation in an Interface Group.

Tip: See "Create Interface Groups" on page 321 for more information about Interface Groups.

For information about each Interface Settings tab:

For information about each Remote NNM 6.x/7.x Events tab:

To apply an incident configuration to a Source Object based on the Source Object's Interface Group:

1. Navigate to the **Remote NNM 6x./7.x Event Configuration** form:
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Expand the **Incidents** folder.
   c. Select **Remote NNM 6.x/7.x Event Configurations**.
   d. Do one of the following:
      i. To create an incident configuration, click the ✱ New icon, and continue.
      ii. To edit an incident configuration, select a row, click the Open icon, and continue.
      iii. To delete an incident configuration, select a row and click the ✖ Delete icon.
2. Select the **Interface Settings** tab.
3. Do one of the following:
   a. To create a new configuration, click the ✱ New icon.
   b. To edit an existing configuration, select a row, click the Open icon, and continue.
4. Configure the desired Interface Settings (see table).
5. Configure any Suppression, Dampen, or Enrichment settings for this Interface Group.
6. Click **Save and Close** to save your changes and return to the Incident Configuration form.

Interface Group Attributes

Name	Description
Interface Group	Click the Lookup icon and select Quick Find to select the Interface Group you want to use. See "Use the Quick Find Window" on page 41 for more information

Interface Group Attributes , continued

Name	Description
	about using Quick Find.
Ordering	Determines the priority order for those interfaces that appear in multiple Interface Groups. The lower the number, the higher the priority. For example, **1** is the highest priority. If an interface is in multiple Interface Groups and more than one of those Interface Groups have been specified in an incident configuration, only the incident configuration with the highest priority will be applied to the interface.
Enable	Use this attribute to temporarily disable an incident's configuration settings. To temporarily disable the Interface Group settings for the selected incident configuration, clear **Enable** □. To enable the Interface Group settings for the selected incident configuration, click **Enable** ☑.

Related Topics

"Configure Node Settings for a Remote NNM 6.x/7.x Event Incident" on page 1278

Configure Incident Suppression Settings for an Interface Group (Remote NNM 6.x/7.x Events)

Note: Interface Settings override any other Suppression settings for this incident, including those from the Node Settings tab.

NNMi enables you to suppress a specified incident configuration based on the Source Object's participation in an Interface Group.

Note: You can also suppress the incident configuration based on the Source Node's participation in a Node Group. See "Configure Incident Suppression Settings for a Node Group (Remote NNM 6.x/7.x Events)" on page 1279 for more information.

Tip: See "Create Interface Groups" on page 321 for more information about Interface Groups.

For information about each Interface Settings tab:

To suppress an incident configuration based on an Interface Group:

1. Navigate to the **Remote NNM 6.x/7.x Event Configuration** form:

   a. From the workspace navigation panel, select the 🔧 **Configuration** workspace.

   b. Click to expand the **Incidents** folder.

   c. Select **Remote NNM 6.x/7.x Event Configurations** .

   d. Do one of the following:

      i. To create an incident configuration, click the ✱ New icon, and continue.

    ii. To edit an incident configuration, select a row, click the Open icon, and continue.

    iii. To delete an incident configuration, select a row and click the Delete icon.

2. Select the **Interface Settings** tab.
3. Do one of the following:
   a. To create a new configuration, click the New icon.

   b. To edit an existing configuration, select a row, click the Open icon, and continue.
4. Make sure the basic Interface Setting behavior is configured. See "Configure Interface Settings for a Remote NNM 6.x/7.x Event Incident" on page 1241 for more information.
5. Select the **Suppression** tab.
6. Configure the desired Suppression behavior (see table).
7. Click **Save and Close** to save your changes and return to the previous form.

Interface Settings Suppression Attributes

Name	Description
Enable	Use this attribute to temporarily disable an incident's suppression settings for the specified Interface Group: **Enable** □ = Temporarily disable the selected configuration. **Enable** ☑ = Enable the selected configuration.
Payload Filter	The Payload Filter Editor enables you to create expressions that further refine the filters used to select the incidents to be suppressed, enriched, or dampened. Make sure to design any complex Payload Filters offline as a Boolean expression first. This method can help to minimize errors when entering your expressions using the Payload Filter editor. When creating a Payload Filter, note the following: • Payload Filter expressions for the `like` and `not like` operators use the syntax defined for java regular expressions (java.util.regex Pattern class). • You must use a `ciaName` that already exists in the trap or event you are configuring. • Each set of expressions associated with a Boolean Operator is treated as if it were enclosed in parentheses and evaluated together. • View the expression displayed under **Filter String** to see the logic of the expression as it is created. • The `AND` and `OR` Boolean Operators must contain at least two expressions as shown in the example below. The following example filters incidents on voltage state: `AND`

Interface Settings Suppression Attributes , continued

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td></td><td>

```
    ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7
    ciaValue = 5
```

NNMi evaluates the expression above as follows:

`(ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7 AND ciaValue = 5)`

NNMi finds all incidents with a varbind `.1.3.6.1.4.1.9.9.13.1.2.1.7` value of **5**.

- The placement of your cursor and the subsequent text that is selected is important when performing operations using the Payload Filter Editor. For example, you append to, replace, or change the indentation of the expression that is selected.
- The placement of your cursor and the subsequent text that is selected is especially important when adding your Boolean operators.
- You can include more than one varbind in the same Payload Filter expression as shown in the following example:

  ```
  ((ciaName like \Q.1.3.6.1.4.1.9.9\E.* AND ciaValue = 25) AND
  (ciaName like \Q.1.3.6.1.2.1.2.2.1.1.3\E.* AND ciaValue = 3))
  ```

  In this example, a given trap must meet each of the following criteria:

  - Contain a varbind whose Object Identifier (OID) matches the regular expression `\Q.1.3.6.1.4.1.9.9\E.*` and has a value of `25`.
  - Contain a varbind whose OID matches the regular expression `\Q.1.3.6.1.2.1.2.2.1.1.3\E.*` and has a value of `3`.

Payload Filter Editor Components

<table>
<tr><th>Attribute</th><th>Description</th></tr>
<tr><td>Attribute</td><td>The attribute name on which NNMi searches. Filterable attributes include the following:

- ciaName
- ciaValue

</td></tr>
<tr><td>Operator</td><td>Valid operators are described below.

- **=** Finds all values equal to the value specified. Click here for an example.

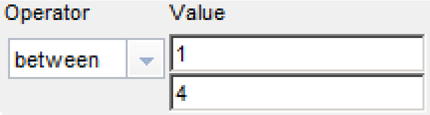
  Example: `ciaName=.1.3.6.1.4.1.9.9.13.1.2.1.7` matches any incident that contains a varbind with the name value of **.1.3.6.1.4.1.9.9.13.1.2.1.7**.
- **!=** Finds all values not equal to the value specified. Click here for an example.

</td></tr>
</table>

</td></tr>
</table>

Interface Settings Suppression Attributes , continued

Name	Description
	Payload Filter Editor Components, continued

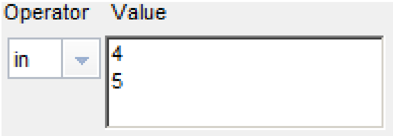
Attribute	Description
	Example: `ciaName! = .1.3.6.1.4.1.9.9.13.1.2.1.7` matches any incident that contains a varbind with a name value other than **1.3.6.1.4.1.9.9.13.1.2.1.7**. • **<** Finds all values less than the value specified. Click here for an example. Example: `ciaValue < 6` matches any incident that contains a varbind with a value less than **6**. • **<=** Finds all values less than or equal to the value specified. Click here for an example. Example: `ciaValue <= 6` matches any incident that contains a varbind with a value less than or equal to **6**. • **>** Finds all values greater than the value specified. Click here for an example. Example: `ciaValue > 4` matches any incident that contains a varbind with a value greater than **4**. • **>=** Finds all values greater than or equal to the value specified. Click here for an example. Example: `ciaValue >= 4` matches any incident that contains a varbind with values greater than or equal to **4**. • **between** Finds all values equal to and between the two values specified. Click here for an example. Example: `ciaValue between` Operator: between Value: 1 4 matches any incident that contains a varbind value equal to or greater than **1** and equal to or less than **4**. **Note**: As shown in the example, each value must be entered on a separate line. • **in** Finds any match to at least one value in a list of values. Click here for an example. Example: `ciaValue in`

Interface Settings Suppression Attributes , continued

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td></td><td>

Payload Filter Editor Components, continued

<table>
<tr><th>Attribute</th><th>Description</th></tr>
<tr><td></td><td>



Operator: in
Value:
4
5

matches any incident with a varbind value of either **4** or **5**.

Note: As shown in the example, each value must be entered on a separate line.

NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example (**4, 5**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line.

- **is not null** Finds all non-blank values. Click here for an example.

  Example: `ciaValue is not null` matches any incident with a varbind that contains a value.

- **is null** Finds all blank values. Click here for an example.

  Example: `ciaValue is null` matches any incident with a varbind that does not contain a value.

- **like** Finds matches using the syntax defined for java regular expressions. See the Pattern (Java Platform SE6) API documentation at : `http://download.oracle.com/javase/6/docs/api/java/util/regex/Pattern.html` for more information. Click here for more information.

  The period asterisk (.*) characters mean *any number of characters of any type at this location*.

  The period (.) character means *any single character of any type at this location*.

  Note: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below.

  Example:

  `ciaName like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that begins with 1.3.6.1.4.1.9.9 and (optionally) ends with any number of characters.

  `ciaValue like .*Chicago.*` finds all traps or events that contain a

</td></tr>
</table>

</td></tr>
</table>

Interface Settings Suppression Attributes , continued

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td></td><td>

Payload Filter Editor Components, continued

<table>
<tr><th>Attribute</th><th>Description</th></tr>
<tr><td></td><td>

varbind value that includes the string **Chicago**.

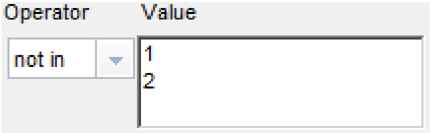
- **not between** Finds all values except those between the two values specified. Click here for an example.

  Example: `ciaValue not between 5 8` matches an incident that contains a varbind with the values less than **5** or greater than **8** .

- **not in** Finds all values except those included in the list of values. Click here for an example.

  Example:

  `ciaValue not in`

  Operator: not in
  Value: 1
2

  matches any incident that contains a varbind with values other than **1** and **2**.

  Note: As shown in the example, each value must be entered on a separate line.

  NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example, (**1, 2**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line.

- **not like** Finds all that do not have the values specified using the syntax defined for Java regular expressions. See the Pattern (Java Platform SE6) API documentation at: `http://download.oracle.com/javase/6/docs/api/java/util/regex/Pattern.html` for more information. Click here for an example.

  The period asterisk (.*) characters mean *any number of characters of any type at this location*.

  The period (.) character means *any single character of any type at this location*.

  Note: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below.

</td></tr>
</table>

</td></tr>
</table>

Interface Settings Suppression Attributes , continued

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td></td><td>
<b>Payload Filter Editor Components, continued</b>
<table>
<tr><th>Attribute</th><th>Description</th></tr>
<tr><td></td><td>Example:
<code>ciaName not like \Q.1.3.6.1.4.1.9.9\E.*</code> matches any incident that contains a varbind name value that does not begin with <b>.1.3.6.1.4.1.9.9</b> and (optionally) ends with any number of characters.
<code>ciaValue not like .*Chicago.*</code> finds all traps or events that do not contain a varbind value that includes the string <b>Chicago</b>.</td></tr>
<tr><td>Value</td><td>The value for which you want NNMi to search.
Note the following:
• The values you enter are case sensitive.
• NNMi displays a variable number of value fields depending on the Operator selected. For example, the <code>between</code> Operator causes two value fields to be displayed.
• The <code>between, in</code> and <code>not in</code> operators require that each value be entered on a separate line.</td></tr>
</table>
<b>Payload Filter Editor Buttons</b>
<table>
<tr><th>Button</th><th>Description</th></tr>
<tr><td>Append</td><td>Appends the current expression (Attribute, Operator,and Value) to the selected expression already included in the filter string.</td></tr>
<tr><td>Insert</td><td>Inserts the current expression (Attribute, Operator,and Value) in front of the cursor location within the Filter String.</td></tr>
<tr><td>Replace</td><td>Replaces the selected expression with the expression displayed in the Attribute, Operator, and Value fields.</td></tr>
<tr><td>AND</td><td>Inserts the AND Boolean Operator in the selected cursor location.
<b>Note</b>: View the expression displayed under <b>Filter String</b> to see the logic of the expression as it is created.</td></tr>
<tr><td>OR</td><td>Inserts the OR Boolean Operator in the current cursor location.
<b>Note</b>: View the expression displayed under <b>Filter String</b> to see the logic of the expression as it is created.</td></tr>
<tr><td>NOT</td><td>Can be used in any part of the Filter String to specify that NNMi should</td></tr>
</table>
</td></tr>
</table>

Interface Settings Suppression Attributes , continued

<table>
<tr><th>Nam e</th><th>Description</th></tr>
<tr><td></td><td>
<b>Payload Filter Editor Buttons, continued</b>
<table>
<tr><th>Button</th><th>Description</th></tr>
<tr><td></td><td>exclude interfaces with values that pass the expression that immediately follows the <code>NOT</code>.
For example, when evaluating the following expression, NNMi includes interfaces with (interface description) <code>ifDesc</code> containing <b>VLAN</b>, and excludes any Interfaces that have <b>VLAN10</b> for the (interface name) <code>ifName</code> value:
<code>(ifDesc like VLAN AND NOT (ifName=VLAN10))</code>
<b>Note</b>: View the expression displayed under Filter String to see the logic of the expression as it is created .</td></tr>
<tr><td>EXISTS</td><td>Used for filters that include Capabilities or Custom Attribute names and values in the Filer String.
Indicates that you want NNMi to consider interfaces that do not have any Capabilities or Custom Attributes when evaluating the Filter String.
<b>Note</b>: If you include Capabilities or Custom Attribute names and values in the Filer String, but do not use <b>EXISTS</b> or <b>NOT EXISTS</b>, NNMi excludes from its search interfaces that do not include Capabilities or Custom Attributes.
For example, when evaluating the following Filter String, NNMi includes interfaces with (interface description) <code>ifDesc</code> containing <b>VLAN</b>, as well as any Interfaces Custom Attribute Role value is <b>LAN Connection to Oracle Server</b>:
<code>(ifDesc like VLAN OR EXISTS((customAttrName=Role AND customAttrValue=LAN Connection to Oracle Server)))</code>
<b>Note</b>: View the expression displayed under Filter String to see the logic of the expression as it is created.</td></tr>
<tr><td>NOT EXISTS</td><td>Used for filters that include Capabilities or Custom Attribute names and values in the Filer String. Indicates that you want NNMi to consider interfaces that do not have any Capabilities or Custom Attributes when evaluating the Filter String, but exclude the interfaces that match the expression that follows the <b>NOT EXISTS</b>.
<b>Note</b>: If you include Capabilities or Custom Attribute names and values in the Filer String, but do not use <b>EXISTS</b> or <b>NOT EXISTS</b>, NNMi excludes from its search interfaces that do not</td></tr>
</table>
</td></tr>
</table>

Interface Settings Suppression Attributes , continued

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td></td><td>

Payload Filter Editor Buttons, continued

<table>
<tr><th>Button</th><th>Description</th></tr>
<tr><td></td><td>include Capabilities or Custom Attributes.
For example, when evaluating the following expression, NNMi includes interfaces with (interface description) <code>ifDesc</code> containing <b>VLAN</b>, and excludes any Interfaces that have the Custom Attribute Role and that Role value is <b>LAN Connection to Oracle Server</b>:
<code>(ifDesc like VLAN OR NOT EXISTS ((customAttrName=Role AND customAttrValue=LAN Connection to Oracle Server)))</code>
<b>Note</b>: View the expression displayed under Filter String to see the logic of the expression as it is created.</td></tr>
<tr><td>Delete</td><td>Deletes the selected expression.
<b>Note</b>: If the Boolean Operator is selected, the Payload Filter Editor deletes all expressions associated with the Boolean Operator.</td></tr>
</table>

</td></tr>
</table>

Configure Incident Enrichment Settings for an Interface Group (Remote NNM 6.x/7.x Events)

Note: Interface Settings override any other Enrichment settings for this incident, including those from the Node Settings tab.

NNMi enables you to fine tune and enhance a specified incident configuration based on the Source Object's participation in an Interface Group. The types of items you can fine tune and enhance for a selected incident configuration, include:

- Category
- Family
- Severity
- Priority
- Correlation Nature
- Message
- Assigned To

Note: You can also enhance the incident configuration based on the Source Node's participation in a Node Group. See "Configure Incident Enrichment Settings for a Node Group (Remote NNM 6.x/7.x Events)" on page 1287 for more information.

Tip: See Create Interface Groups for more information about Interface Groups.

For information about each Interface Settings tab:

For information about each Enrichment tab:

To enrich an incident configuration based on an Interface Group:

1. Navigate to the **Remote NNM 6x./7.x Event Configuration** form:
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Expand the **Incidents** folder.
   c. Select **Remote NNM 6.x/7.x Event Configuratons** .
   d. Do one of the following:
      i. To create an incident configuration, click the ✱ New icon, and continue.
      ii. To edit an incident configuration, select a row, click the Open icon, and continue.
      iii. To delete an incident configuration, select a row and click the ✖ Delete icon.
2. Select the **Interface Settings** tab.
3. Do one of the following:
   a. To create a new configuration, click the ✱ New icon.
   b. To edit an existing configuration, select a row, click the Open icon, and continue.
4. Make sure you configured the basic Interface Setting behavior. See "Configure Interface Settings for a Remote NNM 6.x/7.x Event Incident" on page 1241 for more information.
5. Select the **Enrichment** tab.
6. Do one of the following:
   a. To create an Enrichment configuration, click the ✱ New icon and continue.
   b. To edit an Enrichment configuration, select a row, click the Open icon, and continue.
   c. To delete an Enrichment configuration, select a row and click the ✖ Delete icon.
7. Configure the desired Enrichment behavior (see the "Enrich Configuration Attributes" table)
8. Click **Save and Close** to save your changes and return to the previous form.

Interface Settings Enrichment Attributes

Name	Description
Category	Use the Category attribute to customize the category for this incident configuration. Possible values include: • Accounting • Application Status • Configuration • Fault • Performance

Interface Settings Enrichment Attributes , continued

Name	Description
	- Security - Status See "Specify Category and Family Attribute Values for Organizing Your Incidents (SNMP Trap Incident)" on page 790 for more information.
Family	Use the Family attribute to customize the Family for this incident configuration. Select from the drop-down list or create a new value. For example, some of the values provided by NNMi include: - Address - Aggregated Port (Interfaces using **Link Aggregation**[1] protocol. See Interface Form: Link Aggregation tab.) - Card - Connection - Correlation - Interface - Node
Severity	The incident Severity represents the seriousness calculated for the incident. Use the Severity attribute to specify the severity that should be assigned to the incident you are configuring. Possible values are described below: **Normal** - Indicates there are no known problems related to the associated object. This Severity is meant to be informational. Generally, no action is needed for these incidents. **Warning** - Indicates there might be a problem related to the associated object. **Minor** - Indicates NNMi has detected problems related to the associated object that require further investigation. **Major** - Indicates NNMi has detected problems related to the associated object to be resolved before they become critical. **Critical** - Indicates NNMi has detected problems related to the associated object that require immediate attention.

[1] Protocols used on Switches to configure multiple Interfaces (Aggregation Member Interfaces) to function as if they were one (an Aggregator Interface). When two Aggregator Interfaces establish a connection, that connection is an Aggregator Layer 2 Connection. The Aggregator Layer 2 Connection appears on Layer 2 Neighbor View maps as a thick line with an Interface icon at each end (representing the Aggregator Interface). When you double-click the thick line, it converts into multiple thin lines representing the participating Aggregation Member Layer 2 Connections with their Aggregation Member Interfaces at each end of the lines.

Interface Settings Enrichment Attributes , continued

Name	Description
Priority	Used to communicate the urgency of resolving the selected incident. You control this value. NNMi sets this value to null by default. The lower the number the higher the priority. Possible values are: 5 **None** 4 **Low** 3 **Medium** 2 **High** 1 **Top** **Note**: The icons are displayed only in table views.
Correlation Nature	Use the Correlation Nature to customize the Correlation Nature for this incident configuration. Possible values include: • Info • None • Root Cause • Secondary Root Cause • Symptom • Stream Correlation • Service Impact • Dedup Stream Correlation • Rate Stream Correlation See Incident Form: General Tab for more information.
Message Format	When configuring an incident, specify how the incident message appears in the incident view. The string you specify in the Message Format attribute is visible in an incident view. **Note**: The incident message limit is 1024 characters. If you exceed this limit, NNMi truncates the value starting from the right. You can use any combination of default and custom attributes: "Valid Parameters for Configuring Incident Messages (Remote NNM 6.x/7.x Events)" on page 1233 "Include Custom Incident Attributes in Your Message Format (Remote NNM 6.x/7.x Events)" on page 1239

Interface Settings Enrichment Attributes , continued

Name	Description
Assigned To	Use to specify the owner of any incident generated for this incident configuration. Click the Lookup icon and select Quick Find to select a valid user name. **Note**: You can also begin to type a valid user name and use the auto-complete feature to select the user name of interest.
Description	Use the Description attribute to provide additional information that you want to note about the current enhancement configuration. This description applies only to the enhancement configuration and does not appear when NNMi displays any associated incident. Type a maximum of 1024 characters. Alpha-numeric and special characters (~ ! @ # $ % ^ & * ( ) _+ -) are permitted.

Configure Custom Incident Attributes to Enrich an Incident Configuration (Interface Settings) (Remote NNM 6.x/7.x Events)

The Custom Incident Attributes (CIAs) tab enables you to specify additional CIAs to be saved with an incoming incident. The persisted data might then be used as an argument to an action for the incident.

When creating a CIA for an incident configuration, you can specify any of the following values:

- Custom Attribute on the source node
- Custom Attribute on the interface (source object)

You also specify the Custom Incident Attribute name that will store this information.

For information about each Enrichment tab:

To create a Custom Incident Attribute to enrich an incident configuration:

1. Navigate to the **Remote NNM 6.x/7.x Event Configuration** form:
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Expand the **Incidents** folder.
   c. Select **Remote NNM 6.x/7.x Event Configurations** .
   d. Do one of the following:
      i. To create an incident configuration, click the New icon, and continue.
      ii. To edit an incident configuration, select a row, click the Open icon, and continue.
      iii. To delete an incident configuration, select a row and click the Delete icon.

2. Select the **Interface Settings** tab.
3. Do one of the following:
   a. To create a new configuration, click the New icon.
   b. To edit an existing configuration, select a row, and click the Open icon.
   c. To delete an existing configuration, select a row and click the Delete icon.
4. Make sure you configure the basic Interface Setting behavior. See "Configure Interface Settings for a Remote NNM 6.x/7.x Event Incident" on page 1241 for more information.
5. Select the **Enrichment** tab.
6. Do one of the following:
   a. To create an Enrichment configuration, click the New icon, and continue.
   b. To edit an Enrichment configuration,select a row, click the Open icon, and continue.
   c. To delete an Enrichment configuration, select a row and click the Delete icon.
7. Make sure the Enrichment settings are configured. See "Configure Incident Enrichment Settings for an Interface Group (Remote NNM 6.x/7.x Events)" on page 1250 for more information.
8. Navigate to the **Custom Incident Attributes** tab.
9. Do one of the following:
   a. To create a Custom Incident Attribute, click the New icon, and continue.
   b. To edit a Custom Incident Attribute, select a row, click the Open icon, and continue.
   c. To delete a Custom Incident Attribute, select a row and click the Delete icon.
10. Configure the Custom Incident Attribute (see the "Custom Incident Attribute" table).
11. Click **Save and Close** to save your changes and return to the previous form.

Custom Incident Attribute

Name	Description
Custom Incident Attribute Name	Name used to identify the Custom Incident Attribute you are configuring. The name limit is 255 characters. Alpha-numeric characters are permitted. No spaces or special characters (~ ! @ # $ % ^ & * ( ) _+ -) are permitted.
Type	Specifies whether you are using a Custom Attribute on a node or a Custom Attribute on an interface. Possible values are: • Node Custom Attribute • Interface Custom Attribute
Custom Attribute Name	Used to determine the value to be assigned to the Custom Incident Attribute you are configuring. Enter either of the following: • Name of the Custom Attribute on the source node • Name of the Custom Attribute on the interface (source object)

Configure a Payload Filter to Enrich an Incident Configuration (Interface Settings) (Remote NNM 6.x/7.x Events)

The Payload Filter Editor enables you to create expressions that further refine the filters used to select the incidents to be enriched. Make sure to design any complex Payload Filters offline as a Boolean expression first. This method can help to minimize errors when entering your expressions using the Payload Filter editor.

To create a Payload Filter expression:

1. Navigate to the **Remote NNM 6x./7.x Event Configuration** form:
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Expand the **Incidents** folder.
   c. Select **Remote NNM 6.x/7.x Event Configurations** .
   d. Do one of the following:
      i. To create an incident configuration, click the New icon, and continue.
      ii. To edit an incident configuration, select a row, click the Open icon, and continue.
      iii. To delete an incident configuration, select a row and click the Delete icon.
2. Select the **Interface Settings** tab.
3. Do one of the following:
   a. To create a new configuration, click the New icon.
   b. To edit an existing configuration, select a row, click the Open icon, and continue.
   c. To delete an existing configuration, select a row and click the Delete icon.
4. Make sure you configured the basic Interface Setting behavior. See "Configure Interface Settings for a Remote NNM 6.x/7.x Event Incident" on page 1241 for more information.
5. Select the **Enrichment** tab.
6. Do one of the following:
   a. To create an Enrichment configuration, click the New icon, and continue.
   b. To edit an Enrichment configuration, select a row, click the Open icon, and continue.
   c. To delete an Enrichment configuration, select a row and click the Delete icon.
7. Make sure the Enrichment settings are configured. See "Configure Incident Enrichment Settings for an Interface Group (Remote NNM 6.x/7.x Events)" on page 1250 for more information.
8. Select the **Payload Filter** tab.
9. Define your Payload Filter (see table). Also see "Guidelines for Creating a Payload Filter".

a. Plan out the logic needed for your Filter String.

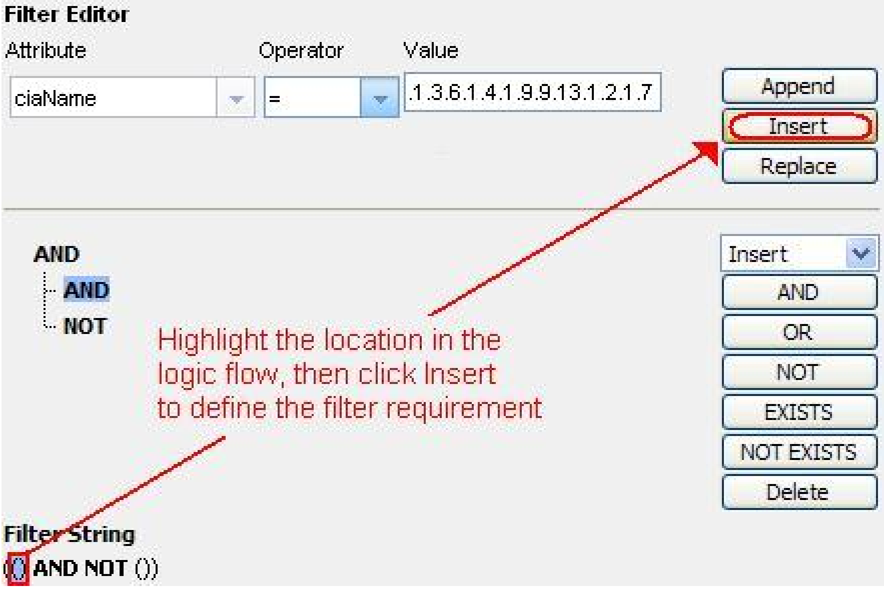
b. Use the buttons on the bottom half of the Additional Filters Editor to establish the logic structure.

   For example, to establish the following structure, select **Insert**, then click **AND**, then **NOT**, and then **AND** a second time:

   ```
   (( ) AND NOT ( ))
   ```

c. Now place your cursor in a location within the displayed Filter String, and use the top half of the filter editor to define the parameters of the highlighted filter requirement.

   For example, select a set of parentheses and use the Insert button to specify the filter requirement within those parentheses:



Filter Editor

Attribute: ciaName
Operator: =
Value: .1.3.6.1.4.1.9.9.13.1.2.1.7

AND
- AND
- NOT

Highlight the location in the logic flow, then click Insert to define the filter requirement

Filter String
(()) AND NOT ())

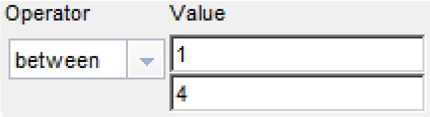
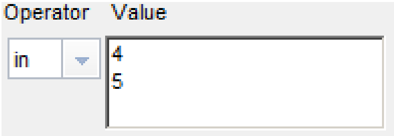
10. Click **Save and Close**.

11. Click **Save and Close** to save your changes and return to the previous form.

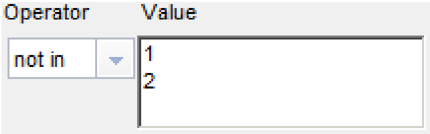
Payload Filter Editor Components

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: • ciaName • ciaValue
Operator	Valid operators are described below. • **=** Finds all values equal to the value specified. Click here for an example. Example: `ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7` matches any incident that contains a varbind with the name value **.1.3.6.1.4.1.9.9.13.1.2.1.7**. • **!=** Finds all values not equal to the value specified. Click here for an example.

Payload Filter Editor Components, continued

Attribute	Description
	Example: `ciaName != .1.3.6.1.4.1.9.9.13.1.2.1.7` matches any incident that contains a varbind with the name value other than **1.3.6.1.4.1.9.9.13.1.2.1.7**. • **<** Finds all values less than the value specified. Click here for an example. Example: `ciaValue < 6` matches any incident with a varbind value less than **6**. • **<=** Finds all values less than or equal to the value specified. Click here for an example. Example: `ciaValue <= 6` matches any incident with a varbind value less than or equal to **6**. • **>** Finds all values greater than the value specified. Click here for an example. Example: `ciaValue > 4` matches any incident with a varbind value greater than **4**. • **>=** Finds all values greater than or equal to the value specified. Click here for an example. Example: `ciaValue >= 4` matches any incident that contains a varbind with values greater than or equal to **4**. • **between** Finds all traps or events that include a varbind with a value equal to and between the two values specified. Click here for an example. Example: `ciaValue between`  Operator Value between 1 4 matches any incident that contains a varbind value equal to or greater than **1** and equal to or less than **4**. **Note**: As shown in the example, each value must be entered on a separate line. • **in** Finds any match to at least one value in a list of values. Click here for an example. Example: `ciaValue in`  Operator Value in 4 5 matches any incident with a varbind value of either **4** or **5**. **Note**: As shown in the example, each value must be entered on a separate line. NNMi displays the list of attributes using comma-separated values enclosed in

Payload Filter Editor Components, continued

Attribute	Description
	parentheses, for example (**4, 5**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line. - **is not null** Finds all non-blank values. Click here for an example. Example: `ciaValue is not null` matches any incident with a varbind that contains a value. - **is null** Finds all blank values. Click here for an example. Example: `ciaValue is null` matches any incident with a varbind that does not have a value. - **like** Finds matches using wildcard characters. Click here for more information about using wildcard characters. The period asterisk (.*) characters mean *any number of characters of any type at this location*. The period (.) character means *any single character of any type at this location*. **Note**: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below. Examples: `ciaName like  \Q.1.3.6.1.4.1.9.9\E.*` finds all traps or events that contain varbind names that begin with **.1.3.6.1.4.1.9.9** and (optionally) end with any number of characters. `ciaValue like .*Chicago.*` finds all traps or events that contain a varbind value that includes the string **Chicago**. - **not between** Finds all values except those between the two values specified. Click here for an example. Example: `ciaValue not between 5 8` matches an incident that contains a varbind with the values less than **5** or greater than **8**. - **not in** Finds all values except those included in the list of values. Click here for an example. Example: `ciaValue not in`  Operator: not in Value: 1 2 matches any incident that contains a varbind with values other than **1** and **2**. **Note**: As shown in the example, each value must be entered on a separate line. NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example, (**1, 2**). However, the comma-separated list is used only

Payload Filter Editor Components, continued

Attribute	Description
	for display purposes. The actual delimiter is the new line. - **not like** Finds all that do not have the values specified (using wildcard strings). Click here for an example. The period asterisk (.*) characters mean *any number of characters of any type at this location*. The period (.) character means *any single character of any type at this location*. **Note**: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below. Example: `ciaName not like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that does not begin with **.1.3.6.1.4.1.9.9**. `ciaValue not like .*Chicago.*` finds all traps or events that do not contain a varbind value that includes the string **Chicago**.
Value	The value for which you want NNMi to search. Note the following: - The values you enter are case sensitive. - NNMi displays a variable number of value fields depending on the Operator selected. For example, the `between` Operator causes two value fields to be displayed. - The `between, in` and `not in` operators require that each value be entered on a separate line.

Additional Filters Editor Buttons

Button	Description
Append	Appends the current expression (Attribute, Operator,and Value) to the selected expression already included in the Filter String.
Insert	Inserts the current expression (Attribute, Operator,and Value) in front of the cursor location within the Filter String.
Replace	Replaces the selected expression with the expression displayed in the Attribute, Operator, and Value fields.
AND	Appends, inserts, or replaces the AND Boolean Operator in the selected cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
OR	Appends, Inserts, or replaces the OR Boolean Operator in the current cursor location.

Additional Filters Editor Buttons, continued

Button	Description
	Note: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
NOT	Can be used in any part of the Filter String to specify that NNMi should exclude interfaces with values that pass the expression that immediately follows the `NOT`. For example, when evaluating the following expression, NNMi includes interfaces with (interface description) `ifDesc` containing **VLAN**, and excludes any Interfaces that have **VLAN10** for the (interface name) `ifName` value: `(ifDesc like VLAN AND NOT (ifName=VLAN10))` **Note**: View the expression displayed under Filter String to see the logic of the expression as it is created .
EXISTS	Used for filters that include Capabilities or Custom Attribute names and values in the Filer String. Indicates that you want NNMi to consider interfaces that do not have any Capabilities or Custom Attributes when evaluating the Filter String. **Note**: If you include Capabilities or Custom Attribute names and values in the Filer String, but do not use **EXISTS** or **NOT EXISTS**, NNMi excludes from its search interfaces that do not include Capabilities or Custom Attributes. For example, when evaluating the following Filter String, NNMi includes interfaces with (interface description) `ifDesc` containing **VLAN**, as well as any Interfaces Custom Attribute Role value is **LAN Connection to Oracle Server**: `(ifDesc like VLAN OR EXISTS((customAttrName=Role AND customAttrValue=LAN Connection to Oracle Server)))` **Note**: View the expression displayed under Filter String to see the logic of the expression as it is created.
NOT EXISTS	Used for filters that include Capabilities or Custom Attribute names and values in the Filer String. Indicates that you want NNMi to consider interfaces that do not have any Capabilities or Custom Attributes when evaluating the Filter String, but exclude the interfaces that match the expression that follows the **NOT EXISTS**. Note: If you include Capabilities or Custom Attribute names and values in the Filer String, but do not use **EXISTS** or **NOT EXISTS**, NNMi excludes from its search interfaces that do not include Capabilities or Custom Attributes. For example, when evaluating the following expression, NNMi includes interfaces with (interface description) `ifDesc` containing **VLAN**, and excludes any Interfaces that have the Custom Attribute Role and that Role value is **LAN Connection to Oracle Server**: `(ifDesc like VLAN OR NOT EXISTS((customAttrName=Role AND customAttrValue=LAN Connection to Oracle Server)))`

Additional Filters Editor Buttons, continued

Button	Description
	Note: View the expression displayed under Filter String to see the logic of the expression as it is created.
Delete	Deletes the selected expression. **Note**: If the Boolean Operator is selected, the Additional Filters Editor deletes all expressions associated with the Boolean Operator.

Configure Incident Dampening Settings for an Interface Group (Remote NNM 6.x/7.x Events)

Note: Interface Settings override any other Dampening settings for this incident, including those from the Node Settings tab.

NNMi enables you to delay the following for an incident configuration based on the Source Object's participation in an Interface Group:

- Execution of Incident Actions
- Execution of Diagnostics (*HP Network Node Manager iSPI Network Engineering Toolset Software \ NNM iSPI NET*)
- Appearance within Incident views in the NNMi Console

Note: You can also suppress the incident configuration based on the Source Node's participation in a Node Group. See "Configure Incident Dampening Settings for a Node Group (Remote NNM 6.x/7.x Events)" on page 1299 for more information.

Tip: See "Create Interface Groups" on page 321 for more information about Interface Groups.

For information about each Interface Settings tab:

When using the Dampening configuration, note the following:

- NNMi initially assigns incidents with Dampening settings configured a Lifecycle State of DAMPENED.
- After the Dampen Interval has passed, NNMi changes the Lifecycle State to REGISTERED.

See About the Incident Lifecycle for more information about Lifecycle State.

To configure Dampening for an incident based on an Interface Group:

1. Navigate to the **Remote NNM 6.x/7.x Event Configuration** form:
   a. From the workspace navigation panel, select the 🔧 **Configuration** workspace.
   b. Expand the **Incidents** folder.
   c. Select **Remote NNM 6.x/7.x Event Configurations** .
   d. Do one of the following:

   i. To create an incident configuration, click the New icon, and continue.

   ii. To edit an incident configuration, select a row, click the Open icon, and continue.

   iii. To delete an incident configuration, select a row and click the Delete icon.

2. Select the **Interface Settings** tab.
3. Do one of the following:
   a. To create a new configuration, click the New icon.

   b. To edit an existing configuration, select a row, click the Open icon, and continue.
4. Make sure you configured the basic Interface Setting behavior. See "Configure Interface Settings for a Remote NNM 6.x/7.x Event Incident" on page 1241 for more information.
5. Select the **Dampening** tab.
6. Configure the desired Dampening behavior (see table).
7. Click **Save and Close** to save your changes and return to the previous form.

Interface Settings Dampening Configuration Attributes

Name	Description
Enable	Use this attribute to temporarily disable an incident's dampening settings: **Enable** □ = Temporarily disable the selected configuration. **Enable** ☑ = Enable the selected configuration.
Hour	Specifies the number of hours to be used for the dampen interval.
Minutes	Specifies the number of minutes to be used for the dampen interval. **Note:** The maximum dampen interval is 60 minutes.
Seconds	Specifies the number of seconds to be used for the dampen interval.
Payload Filter	The Payload Filter Editor enables you to create expressions that further refine the filters used to select the incidents to be suppressed, enriched, or dampened. Make sure to design any complex Payload Filters offline as a Boolean expression first. This method can help to minimize errors when entering your expressions using the Payload Filter editor. When creating a Payload Filter, note the following: • Payload Filter expressions for the `like` and `not like` operators use the syntax defined for java regular expressions (java.util.regex Pattern class). • You must use a `ciaName` that already exists in the trap or event you are configuring. • Each set of expressions associated with a Boolean Operator is treated as if it were enclosed in parentheses and evaluated together. • View the expression displayed under **Filter String** to see the logic of the expression

Interface Settings Dampening Configuration Attributes , continued

Name	Description

	as it is created.
	• The `AND` and `OR` Boolean Operators must contain at least two expressions as shown in the example below.
	The following example filters incidents on voltage state:
	`AND`
	`    ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7`
	`    ciaValue = 5`
	NNMi evaluates the expression above as follows:
	`(ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7 AND ciaValue = 5)`
	NNMi finds all incidents with a varbind `.1.3.6.1.4.1.9.9.13.1.2.1.7` value of **5**.
	• The placement of your cursor and the subsequent text that is selected is important when performing operations using the Payload Filter Editor. For example, you append to, replace, or change the indentation of the expression that is selected.
	• The placement of your cursor and the subsequent text that is selected is especially important when adding your Boolean operators.
	• You can include more than one varbind in the same Payload Filter expression as shown in the following example:
	`((ciaName like \Q.1.3.6.1.4.1.9.9\E.* AND ciaValue = 25) AND (ciaName like \Q.1.3.6.1.2.1.2.2.1.1.3\E.* AND ciaValue = 3))`
	In this example, a given trap must meet each of the following criteria:
	▪ Contain a varbind whose Object Identifier (OID) matches the regular expression `\Q.1.3.6.1.4.1.9.9\E.*` and has a value of `25`.
	▪ Contain a varbind whose OID matches the regular expression `\Q.1.3.6.1.2.1.2.2.1.1.3\E.*` and has a value of `3`.

Payload Filter Editor Components

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: • ciaName • ciaValue
Operator	Valid operators are described below. • **=** Finds all values equal to the value specified. Click here for an example.

Interface Settings Dampening Configuration Attributes , continued

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td></td><td>

Payload Filter Editor Components, continued

<table>
<tr><th>Attribute</th><th>Description</th></tr>
<tr><td></td><td>

Example: `ciaName=.1.3.6.1.4.1.9.9.13.1.2.1.7` matches any incident that contains a varbind with the name value of **.1.3.6.1.4.1.9.9.13.1.2.1.7**.

- **!=** Finds all values not equal to the value specified. Click here for an example.

  Example: `ciaName! = .1.3.6.1.4.1.9.9.13.1.2.1.7` matches any incident that contains a varbind with a name value other than **1.3.6.1.4.1.9.9.13.1.2.1.7**.

- **<** Finds all values less than the value specified. Click here for an example.

  Example: `ciaValue < 6` matches any incident that contains a varbind with a value less than **6**.

- **<=** Finds all values less than or equal to the value specified. Click here for an example.

  Example: `ciaValue <= 6` matches any incident that contains a varbind with a value less than or equal to **6**.

- **>** Finds all values greater than the value specified. Click here for an example.

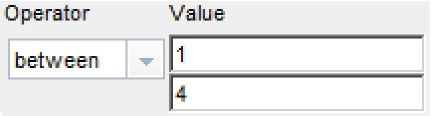
  Example: `ciaValue > 4` matches any incident that contains a varbind with a value greater than **4**.

- **>=** Finds all values greater than or equal to the value specified. Click here for an example.

  Example: `ciaValue >= 4` matches any incident that contains a varbind with values greater than or equal to **4**.

- **between** Finds all values equal to and between the two values specified. Click here for an example.

  Example: `ciaValue between`

  Operator: between
  Value: 1
  4

  matches any incident that contains a varbind value equal to or greater than **1** and equal to or less than **4**.

  Note: As shown in the example, each value must be entered on a separate line.

- **in** Finds any match to at least one value in a list of values. Click here for an example.

</td></tr>
</table>

</td></tr>
</table>

Interface Settings Dampening Configuration Attributes , continued

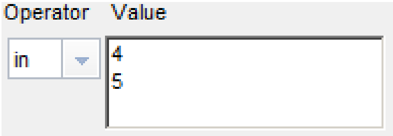
<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td></td><td>

Payload Filter Editor Components, continued

<table>
<tr><th>Attribute</th><th>Description</th></tr>
<tr><td></td><td>

Example:

`ciaValue in`



Operator: in

Value:
4
5

matches any incident with a varbind value of either **4** or **5**.

Note: As shown in the example, each value must be entered on a separate line.

NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example (**4, 5**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line.

- **is not null** Finds all non-blank values. Click here for an example.

  Example: `ciaValue is not null` matches any incident with a varbind that contains a value.

- **is null** Finds all blank values. Click here for an example.

  Example: `ciaValue is null` matches any incident with a varbind that does not contain a value.

- **like** Finds matches using the syntax defined for java regular expressions. See the Pattern (Java Platform SE6) API documentation at : `http://download.oracle.com/javase/6/docs/api/java/util/regex/Pattern.html` for more information. Click here for more information.

  The period asterisk (.*) characters mean *any number of characters of any type at this location*.

  The period (.) character means *any single character of any type at this location*.

  Note: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below.

  Example:

  `ciaName like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that begins with 1.3.6.1.4.1.9.9 and

</td></tr>
</table>

</td></tr>
</table>

Interface Settings Dampening Configuration Attributes , continued

Name	Description

Payload Filter Editor Components, continued

Attribute	Description

(optionally) ends with any number of characters.

`ciaValue like .*Chicago.*` finds all traps or events that contain a varbind value that includes the string **Chicago**.

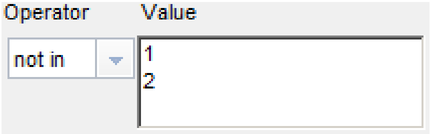
- **not between** Finds all values except those between the two values specified. Click here for an example.

  Example: `ciaValue not between 5 8` matches an incident that contains a varbind with the values less than **5** or greater than **8** .

- **not in** Finds all values except those included in the list of values. Click here for an example.

  Example:

  `ciaValue not in`

  Operator: not in

  Value:
  1
  2

  matches any incident that contains a varbind with values other than **1** and **2**.

  Note: As shown in the example, each value must be entered on a separate line.

  NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example, (**1, 2**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line.

- **not like** Finds all that do not have the values specified using the syntax defined for Java regular expressions. See the Pattern (Java Platform SE6) API documentation at: `http://download.oracle.com/javase/6/docs/api/java/util/regex/Pattern.html` for more information. Click here for an example.

  The period asterisk (.*) characters mean *any number of characters of any type at this location.*

  The period (.) character means *any single character of any type at this location.*

  Note: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed

Interface Settings Dampening Configuration Attributes , continued

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td></td><td>

<b>Payload Filter Editor Components, continued</b>

<table>
<tr><th>Attrib ute</th><th>Description</th></tr>
<tr><td></td><td>below.
Example:
<code>ciaName not like \Q.1.3.6.1.4.1.9.9\E.*</code> matches any incident that contains a varbind name value that does not begin with <b>.1.3.6.1.4.1.9.9</b> and (optionally) ends with any number of characters.
<code>ciaValue not like .*Chicago.*</code> finds all traps or events that do not contain a varbind value that includes the string <b>Chicago</b>.</td></tr>
<tr><td>Value</td><td>The value for which you want NNMi to search.
Note the following:<ul><li>The values you enter are case sensitive.</li><li>NNMi displays a variable number of value fields depending on the Operator selected. For example, the <code>between</code> Operator causes two value fields to be displayed.</li><li>The <code>between, in</code> and <code>not in</code> operators require that each value be entered on a separate line.</li></ul></td></tr>
</table>

<b>Payload Filter Editor Buttons</b>

<table>
<tr><th>Button</th><th>Description</th></tr>
<tr><td>Append</td><td>Appends the current expression (Attribute, Operator,and Value) to the selected expression already included in the filter string.</td></tr>
<tr><td>Insert</td><td>Inserts the current expression (Attribute, Operator,and Value) in front of the cursor location within the Filter String.</td></tr>
<tr><td>Replace</td><td>Replaces the selected expression with the expression displayed in the Attribute, Operator, and Value fields.</td></tr>
<tr><td>AND</td><td>Inserts the AND Boolean Operator in the selected cursor location.
<b>Note</b>: View the expression displayed under <b>Filter String</b> to see the logic of the expression as it is created.</td></tr>
<tr><td>OR</td><td>Inserts the OR Boolean Operator in the current cursor location.
<b>Note</b>: View the expression displayed under <b>Filter String</b> to see the logic of the expression as it is created.</td></tr>
<tr><td>NOT</td><td>Can be used in any part of the Filter String to specify that NNMi</td></tr>
</table>

</td></tr>
</table>

Interface Settings Dampening Configuration Attributes , continued

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td></td><td>

Payload Filter Editor Buttons, continued

<table>
<tr><th>Button</th><th>Description</th></tr>
<tr><td></td><td>should exclude interfaces with values that pass the expression that immediately follows the `NOT`.
For example, when evaluating the following expression, NNMi includes interfaces with (interface description) `ifDesc` containing **VLAN**, and excludes any Interfaces that have **VLAN10** for the (interface name) `ifName` value:
`(ifDesc like VLAN AND NOT (ifName=VLAN10))`
Note: View the expression displayed under Filter String to see the logic of the expression as it is created .</td></tr>
<tr><td>EXISTS</td><td>Used for filters that include Capabilities or Custom Attribute names and values in the Filer String.
Indicates that you want NNMi to consider interfaces that do not have any Capabilities or Custom Attributes when evaluating the Filter String.
Note: If you include Capabilities or Custom Attribute names and values in the Filer String, but do not use **EXISTS** or **NOT EXISTS**, NNMi excludes from its search interfaces that do not include Capabilities or Custom Attributes.
For example, when evaluating the following Filter String, NNMi includes interfaces with (interface description) `ifDesc` containing **VLAN**, as well as any Interfaces Custom Attribute Role value is **LAN Connection to Oracle Server**:
`(ifDesc like VLAN OR EXISTS((customAttrName=Role AND customAttrValue=LAN Connection to Oracle Server)))`
Note: View the expression displayed under Filter String to see the logic of the expression as it is created.</td></tr>
<tr><td>NOT EXISTS</td><td>Used for filters that include Capabilities or Custom Attribute names and values in the Filer String. Indicates that you want NNMi to consider interfaces that do not have any Capabilities or Custom Attributes when evaluating the Filter String, but exclude the interfaces that match the expression that follows the **NOT EXISTS**.
Note: If you include Capabilities or Custom Attribute names and values in the Filer String, but do not use **EXISTS** or **NOT EXISTS**, NNMi excludes from its search interfaces that do not include Capabilities or Custom Attributes.</td></tr>
</table>

</td></tr>
</table>

Interface Settings Dampening Configuration Attributes , continued

Name	Description
	Payload Filter Editor Buttons, continued

Button	Description
	For example, when evaluating the following expression, NNMi includes interfaces with (interface description) `ifDesc` containing **VLAN**, and excludes any Interfaces that have the Custom Attribute Role and that Role value is **LAN Connection to Oracle Server**: `(ifDesc like VLAN OR NOT EXISTS ((customAttrName=Role AND customAttrValue=LAN Connection to Oracle Server)))` **Note**: View the expression displayed under Filter String to see the logic of the expression as it is created.
Delete	Deletes the selected expression. **Note**: If the Boolean Operator is selected, the Payload Filter Editor deletes all expressions associated with the Boolean Operator.

Configure Incident Actions for an Interface Group (Remote NNM 6.x/7.x Event)

Note: Interface Settings override any other Actions settings for this incident, including those from the Node Settings tab.

For information about each Interface Settings tab:

NNMi enables you to configure incident actions based on a Source Object's participation in an Interface Group.

You can configure actions to automatically run at any point in the incident lifecycle. For example, you might want to configure an action to occur when an incident of the type you are configuring is generated (**Registered**). When an incident is generated, you might want to automatically open a trouble ticket or send email or page your network operator. After the incident is **Closed**, you might want to automatically close the trouble ticket.

Note: Your actions will not be executed until you enable the Actions configuration by either clicking Enable ☑ on the Actions tab or using the **Actions** → **Enable Configuration** option.

You can configure actions for incidents generated from SNMP traps, NNM 6.x or 7.x events, and the NNMi management events. Any time an incident configuration changes, the action directory is rescanned and any Jython files are reloaded to the NNMi database. See "Lifecycle Transition Action Form (Remote NNM 6.x/7.x Events)" on page 1358 for more information about the actions directory.

Tip: Copy any required Jython files to the NNMi actions directory before you configure an incident action. New or updated actions are loaded into NNMi only when an incident configuration is updated or created. See "Lifecycle Transition Action Form (Remote NNM 6.x/7.x Events)" on page 1358 for the location of the NNMi action directory.

When the action runs, output is logged to the `incidentActions.*.*.log` file. See "Verify that NNMi Services are Running" on page 85 for more information about log files and where they are located.

To configure an automatic action for an incident:

1. Navigate to the **Remote NNM 6x./7.x Event Configuration** form.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Expand the **Incidents** folder.
   c. Select **Remote NNM 6.x/7.x Event Configurations**.
   d. Do one of the following:
      i. To create a new incident configuration, click the New icon.
      ii. To edit an existing incident configuration, select a row, click the Open icon, and continue.
2. Select the **Interface Settings** tab.
3. Do one of the following:
   a. To create a new configuration, click the New icon.
   b. To edit an existing configuration, select a row, click the Open icon, and continue.
   c. To delete an existing configuration, select a row and click the Delete icon.
4. Make sure you configured the basic Interface Setting behavior. See "Configure Interface Settings for a Remote NNM 6.x/7.x Event Incident" on page 1241 for more information.
5. Select the **Actions** tab.
6. From the **Lifecycle Actions** table toolbar, do one of the following:
   - To create an Action configuration, click the New icon, and continue.
   - To edit an Action configuration, select a row, click the Open icon, and continue.
   - To delete an Action configuration, select a row and click the Delete icon.
7. In the "Lifecycle Transition Action Form (Remote NNM 6.x/7.x Events)" on page 1358, provide the required information.
8. Click **Save and Close** to save your changes and return to the previous form.

   The next time the lifecycle changes, NNMi launches the action associated with the lifecycle for the incident of that type.

Configure a Payload Filter for an Incident Action (Interface Settings) (Remote NNM 6.x/7.x Events)

The Payload Filter Editor enables you to create expressions that further refine the filters used to select the incidents that cause the configured action to run. Make sure to design any complex

Payload Filters offline as a Boolean expression first. This method can help to minimize errors when entering your expressions using the Payload Filter editor.

To create a Payload Filter expression:

1. Navigate to the **Remote NNM 6.x/7.x Event Configuration** form:

   a. From the workspace navigation panel, select the **Configuration** workspace.

   b. Expand the **Incidents** folder.

   c. Select **Remote NNM 6.x/7.x Event Configurations**.

   d. Do one of the following:

      i. To create an incident configuration, click the ✳ New icon, and continue.

      ii. To edit an incident configuration, select a row, click the 📂 Open icon, and continue.

      iii. To delete an incident configuration, select a row and click the ✖ Delete icon.

2. Select the **Interface Settings** tab.

3. Do one of the following:

   a. To create a new configuration, click the ✳ New icon.

   b. To edit a configuration, select a row, click the 📂 Open icon, and continue.

   c. To delete an existing configuration, select a row and click the ✖ Delete icon.

4. Make sure you configure the basic Interface Setting behavior. See "Configure Interface Settings for a Remote NNM 6.x/7.x Event Incident" on page 1241 for more information.

5. Select the **Actions** tab.

6. Do one of the following:

   a. To create an Action configuration, click the ✳ New icon, and continue.

   b. To edit an Action configuration, select a row, click the 📂 Open icon, and continue.

   c. To delete an Action configuration, select a row, and click the ✖ Delete icon.

7. Make sure the Action Configuration settings are configured. See "Configure Incident Actions for an Interface Group (Remote NNM 6.x/7.x Event)" on page 1270 for more information.

8. Select the **Payload Filter** tab.

9. Define your Payload Filter (see table). Also see "Guidelines for Creating a Payload Filter".

   a. Plan out the logic needed for your Filter String.

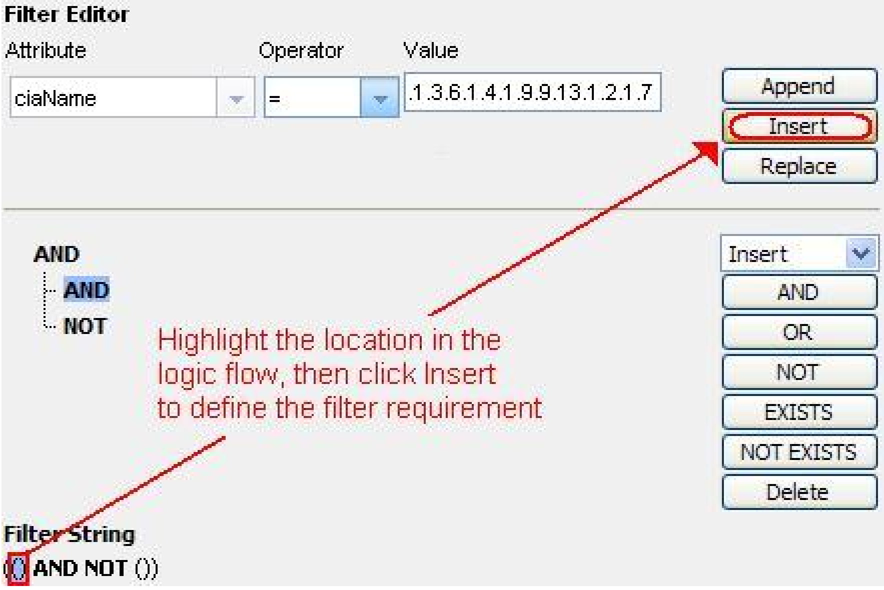
   b. Use the buttons on the bottom half of the Additional Filters Editor to establish the logic structure.

      For example, to establish the following structure, select **Insert**, then click **AND**, then **NOT**, and then **AND** a second time:

      ```
      (( ) AND NOT ( ))
      ```

   c. Now place your cursor in a location within the displayed Filter String, and use the top half of the filter editor to define the parameters of the highlighted filter requirement.

For example, select a set of parentheses and use the Insert button to specify the filter requirement within those parentheses:

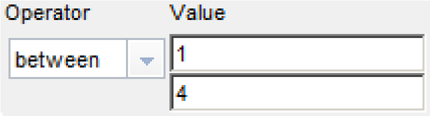
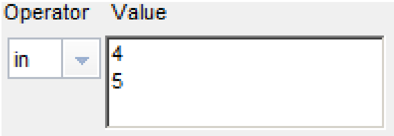


10. Click **Save and Close**.
11. Click **Save and Close** to save your changes and return to the previous form.

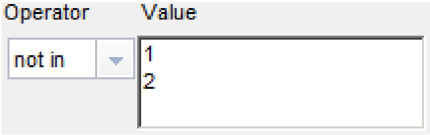
Payload Filter Editor Components

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: • ciaName • ciaValue
Operator	Valid operators are described below. • **=** Finds all values equal to the value specified. Click here for an example. Example: `ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7` matches any incident that contains a varbind with the name value **.1.3.6.1.4.1.9.9.13.1.2.1.7**. • **!=** Finds all values not equal to the value specified. Click here for an example. Example: `ciaName != .1.3.6.1.4.1.9.9.13.1.2.1.7` matches any incident that contains a varbind with the name value other than **1.3.6.1.4.1.9.9.13.1.2.1.7**. • **<** Finds all values less than the value specified. Click here for an example. Example: `ciaValue < 6` matches any incident with a varbind value less than **6**. • **<=** Finds all values less than or equal to the value specified. Click here for an example.

Payload Filter Editor Components, continued

Attribute	Description
	Example: `ciaValue <= 6` matches any incident with a varbind value less than or equal to **6**. • **>** Finds all values greater than the value specified. Click here for an example. Example: `ciaValue > 4` matches any incident with a varbind value greater than **4**. • **>=** Finds all values greater than or equal to the value specified. Click here for an example. Example: `ciaValue >= 4` matches any incident that contains a varbind with values greater than or equal to **4**. • **between** Finds all traps or events that include a varbind with a value equal to and between the two values specified. Click here for an example. Example: `ciaValue between`  Operator Value between 1 4 matches any incident that contains a varbind value equal to or greater than **1** and equal to or less than **4**. **Note**: As shown in the example, each value must be entered on a separate line. • **in** Finds any match to at least one value in a list of values. Click here for an example. Example: `ciaValue in`  Operator Value in 4 5 matches any incident with a varbind value of either **4** or **5**. **Note**: As shown in the example, each value must be entered on a separate line. NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example (**4, 5**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line. • **is not null** Finds all non-blank values. Click here for an example. Example: `ciaValue is not null` matches any incident with a varbind that contains a value. • **is null** Finds all blank values. Click here for an example. Example: `ciaValue is null` matches any incident with a varbind that does

Payload Filter Editor Components, continued

Attribute	Description
	not have a value. • **like** Finds matches using wildcard characters. Click here for more information about using wildcard characters. The period asterisk (.*) characters mean *any number of characters of any type at this location*. The period (.) character means *any single character of any type at this location*. **Note**: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below. Examples: `ciaName like  \Q.1.3.6.1.4.1.9.9\E.*` finds all traps or events that contain varbind names that begin with **.1.3.6.1.4.1.9.9** and (optionally) end with any number of characters. `ciaValue like .*Chicago.*` finds all traps or events that contain a varbind value that includes the string **Chicago**. • **not between** Finds all values except those between the two values specified. Click here for an example. Example: `ciaValue not between 5 8` matches an incident that contains a varbind with the values less than **5** or greater than **8**. • **not in** Finds all values except those included in the list of values. Click here for an example. Example: `ciaValue not in`  Operator: not in Value: 1 2 matches any incident that contains a varbind with values other than **1** and **2**. **Note**: As shown in the example, each value must be entered on a separate line. NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example, (**1, 2**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line. • **not like** Finds all that do not have the values specified (using wildcard strings). Click here for an example. The period asterisk (.*) characters mean *any number of characters of any type at this location*. The period (.) character means *any single character of any type at this location*. **Note**: To include literal string values in the Value attribute, enclose the string value

Payload Filter Editor Components, continued

Attribute	Description
	in \Q<*literal_value*>\E as shown in the Examples listed below. Example: `ciaName not like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that does not begin with **.1.3.6.1.4.1.9.9**. `ciaValue not like .*Chicago.*` finds all traps or events that do not contain a varbind value that includes the string **Chicago**.
Value	The value for which you want NNMi to search. Note the following: • The values you enter are case sensitive. • NNMi displays a variable number of value fields depending on the Operator selected. For example, the `between` Operator causes two value fields to be displayed. • The `between, in` and `not in` operators require that each value be entered on a separate line.

Additional Filters Editor Buttons

Button	Description
Append	Appends the current expression (Attribute, Operator,and Value) to the selected expression already included in the Filter String.
Insert	Inserts the current expression (Attribute, Operator,and Value) in front of the cursor location within the Filter String.
Replace	Replaces the selected expression with the expression displayed in the Attribute, Operator, and Value fields.
AND	Appends, inserts, or replaces the AND Boolean Operator in the selected cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
OR	Appends, Inserts, or replaces the OR Boolean Operator in the current cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
NOT	Can be used in any part of the Filter String to specify that NNMi should exclude interfaces with values that pass the expression that immediately follows the `NOT`. For example, when evaluating the following expression, NNMi includes interfaces with (interface description) `ifDesc` containing **VLAN**, and excludes

Additional Filters Editor Buttons, continued

Button	Description
	any Interfaces that have **VLAN10** for the (interface name) `ifName` value: `(ifDesc like VLAN AND NOT (ifName=VLAN10))` **Note**: View the expression displayed under Filter String to see the logic of the expression as it is created .
EXISTS	Used for filters that include Capabilities or Custom Attribute names and values in the Filer String. Indicates that you want NNMi to consider interfaces that do not have any Capabilities or Custom Attributes when evaluating the Filter String. **Note**: If you include Capabilities or Custom Attribute names and values in the Filer String, but do not use **EXISTS** or **NOT EXISTS**, NNMi excludes from its search interfaces that do not include Capabilities or Custom Attributes. For example, when evaluating the following Filter String, NNMi includes interfaces with (interface description) `ifDesc` containing **VLAN**, as well as any Interfaces Custom Attribute Role value is **LAN Connection to Oracle Server**: `(ifDesc like VLAN OR EXISTS((customAttrName=Role AND customAttrValue=LAN Connection to Oracle Server)))` **Note**: View the expression displayed under Filter String to see the logic of the expression as it is created.
NOT EXISTS	Used for filters that include Capabilities or Custom Attribute names and values in the Filer String. Indicates that you want NNMi to consider interfaces that do not have any Capabilities or Custom Attributes when evaluating the Filter String, but exclude the interfaces that match the expression that follows the **NOT EXISTS**. Note: If you include Capabilities or Custom Attribute names and values in the Filer String, but do not use **EXISTS** or **NOT EXISTS**, NNMi excludes from its search interfaces that do not include Capabilities or Custom Attributes. For example, when evaluating the following expression, NNMi includes interfaces with (interface description) `ifDesc` containing **VLAN**, and excludes any Interfaces that have the Custom Attribute Role and that Role value is **LAN Connection to Oracle Server**: `(ifDesc like VLAN OR NOT EXISTS((customAttrName=Role AND customAttrValue=LAN Connection to Oracle Server)))` **Note**: View the expression displayed under Filter String to see the logic of the expression as it is created.
Delete	Deletes the selected expression. **Note**: If the Boolean Operator is selected, the Additional Filters Editor deletes all expressions associated with the Boolean Operator.

Configure Node Settings for a Remote NNM 6.x/7.x Event Incident

Note: Node Settings override any other Suppression, Enrichment, Dampen, Actions, or Diagnostics Selections configuration settings for this incident, except those configured on the Interface Settings tab.

NNMi enables you to apply an incident configuration to a Source Node based on the Source Node's participation in a Node Group. If the Source Node is not a member of the Node Group specified, the incident is neither displayed nor stored in the NNMi database.

Tip: See "Create Node Groups" on page 295 for more information about Node Groups.

For information about each Node Settings tab:

For information about each Remote NNM 6.x/7.x Events tab:

To apply an incident configuration to a Source Node based on the Source Node's Node Group:

1. Navigate to the **Remote NNM 6.x/7.x Event Configuration** form:
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Expand the **Incidents** folder.
   c. Select **Remote NNM 6.x/7.x Event Configurations** .
   d. Do one of the following:
      i. To create an incident configuration, click the New icon, and continue.
      ii. To edit an incident configuration, select a row, click the Open icon, and continue.
      iii. To delete an incident configuration, select a row and click the Delete icon.
2. Select the **Node Settings** tab.
3. Do one of the following:
   a. To create a new configuration, click the New icon.
   b. To edit an existing configuration,select a row, click the Open icon, and continue.
4. Configure the desired Node Settings (see table).
5. Click **Save and Close** to save your changes and return to the previous form.

Node Group Attributes

Name	Description
Node Group	Click the Lookup icon and select Quick Find to select the Node Group you want to use. See "Use the Quick Find Window" on page 41 for more information about using Quick Find.
Ordering	Determines the priority order for those nodes that appear in multiple Node Groups. The lower the number, the higher the priority. For example, **1** is the highest priority. If a

Node Group Attributes , continued

Name	Description
	node is in multiple Node Groups and more than one of those Node Groups have been specified in an incident configuration, only the incident configuration with the highest priority will be applied to the node.
Enable	Use this attribute to temporarily disable an incident's suppression settings. To temporarily disable the Node Group settings for the selected incident configuration, clear **Enable** □. To enable the Node Group settings for the selected incident configuration, click **Enable** ☑.

Configure Incident Suppression Settings for a Node Group (Remote NNM 6.x/7.x Events)

Note: Node Settings override any other Suppression settings for this incident, except those configured on the Interface Settings tab.

NNMi enables you to suppress a specified incident configuration based on the Source Node's participation in a Node Group.

Note: You can also suppress the incident configuration based on the Source Object's participation in an Interface Group. See "Configure Incident Suppression Settings for an Interface Group (SNMP Trap Incident)" on page 804 for more information.

Tip: See "Create Node Groups" on page 295 for more information about Node Groups.

For information about each Node Settings tab:

To suppress an incident configuration based on a Node Group:

1. Navigate to the **Incident Configuration** form:
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Expand the **Incidents** folder.
   c. Select **Remote NNM 6.x/7.x Event Configurations**.
   d. Do one of the following:
      i. To create an incident configuration, click the New icon, and continue.
      ii. To edit an incident configuration, select a row, click the Open icon, and continue.
      iii. To delete an incident configuration, select a row and click the Delete icon.
2. Select the **Node Settings** tab.
3. Do one of the following:
   a. To create a new configuration, click the New icon.
   b. To edit an existing configuration, select a row, click the Open icon, and continue.

c. To delete an existing configuration, select a row and click the ✖ Delete icon.

4. Make sure you configured the basic Node Setting behavior. See "Configure Node Settings for a Remote NNM 6.x/7.x Event Incident" on page 1278 for more information.
5. Select the **Suppression** tab.
6. Configure the desired Suppression behavior (see table).
7. Click **Save and Close** to save your changes and return to the previous form.

Node Settings Suppression Attributes

Name	Description
Enable	Use this attribute to temporarily disable an incident's suppression settings: **Enable** □ = Temporarily disable the selected configuration. **Enable** ☑ = Enable the selected configuration.

Payload Filter	The Payload Filter Editor enables you to create expressions that further refine the filters used to select the incidents to be suppressed, enriched, or dampened. Make sure to design any complex Payload Filters offline as a Boolean expression first. This method can help to minimize errors when entering your expressions using the Payload Filter editor.
	When creating a Payload Filter, note the following:
	• Payload Filter expressions for the `like` and `not like` operators use the syntax defined for java regular expressions (java.util.regex Pattern class).
	• You must use a `ciaName` that already exists in the trap or event you are configuring.
	• Each set of expressions associated with a Boolean Operator is treated as if it were enclosed in parentheses and evaluated together.
	• View the expression displayed under **Filter String** to see the logic of the expression as it is created.
	• The `AND` and `OR` Boolean Operators must contain at least two expressions as shown in the example below.
	The following example filters incidents on voltage state:
	`AND`
	    `ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7`
	    `ciaValue = 5`
	NNMi evaluates the expression above as follows:
	`(ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7 AND ciaValue = 5)`
	NNMi finds all incidents with a varbind `.1.3.6.1.4.1.9.9.13.1.2.1.7` value of **5**.
	• The placement of your cursor and the subsequent text that is selected is important when performing operations using the Payload Filter Editor. For example, you append

Node Settings Suppression Attributes , continued

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td></td><td>to, replace, or change the indentation of the expression that is selected.
- The placement of your cursor and the subsequent text that is selected is especially important when adding your Boolean operators.
- You can include more than one varbind in the same Payload Filter expression as shown in the following example:
`((ciaName like \Q.1.3.6.1.4.1.9.9\E.* AND ciaValue = 25) AND (ciaName like \Q.1.3.6.1.2.1.2.2.1.1.3\E.* AND ciaValue = 3))`
In this example, a given trap must meet each of the following criteria:
▪ Contain a varbind whose Object Identifier (OID) matches the regular expression `\Q.1.3.6.1.4.1.9.9\E.*` and has a value of `25`.
▪ Contain a varbind whose OID matches the regular expression `\Q.1.3.6.1.2.1.2.2.1.1.3\E.*` and has a value of `3`.
Payload Filter Editor Components
<table>
<tr><th>Attribute</th><th>Description</th></tr>
<tr><td>Attribute</td><td>The attribute name on which NNMi searches. Filterable attributes include the following:
- ciaName
- ciaValue</td></tr>
<tr><td>Operator</td><td>Valid operators are described below.
- **=** Finds all values equal to the value specified. Click here for an example.
Example: `ciaName=.1.3.6.1.4.1.9.9.13.1.2.1.7` matches any incident that contains a varbind with the name value of **.1.3.6.1.4.1.9.9.13.1.2.1.7**.
- **!=** Finds all values not equal to the value specified. Click here for an example.
Example: `ciaName! = .1.3.6.1.4.1.9.9.13.1.2.1.7` matches any incident that contains a varbind with a name value other than **1.3.6.1.4.1.9.9.13.1.2.1.7**.
- **<** Finds all values less than the value specified. Click here for an example.
Example: `ciaValue < 6` matches any incident that contains a varbind with a value less than **6**.
- **<=** Finds all values less than or equal to the value specified. Click here for an example.</td></tr>
</table>
</td></tr>
</table>

Node Settings Suppression Attributes , continued

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td></td><td>

Payload Filter Editor Components, continued

<table>
<tr><th>Attribute</th><th>Description</th></tr>
<tr><td></td><td>

Example: `ciaValue <= 6` matches any incident that contains a varbind with a value less than or equal to **6**.

- **>** Finds all values greater than the value specified. Click here for an example.

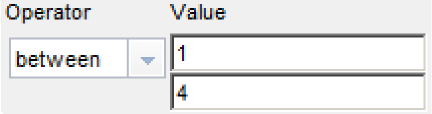
  Example: `ciaValue > 4` matches any incident that contains a varbind with a value greater than **4**.

- **>=** Finds all values greater than or equal to the value specified. Click here for an example.

  Example: `ciaValue >= 4` matches any incident that contains a varbind with values greater than or equal to **4**.

- **between** Finds all values equal to and between the two values specified. Click here for an example.

  Example: `ciaValue between`

  Operator: between

  Value: 1 4

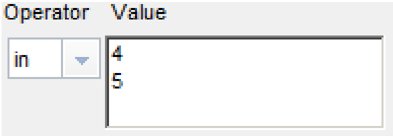
  matches any incident that contains a varbind value equal to or greater than **1** and equal to or less than **4**.

  Note: As shown in the example, each value must be entered on a separate line.

- **in** Finds any match to at least one value in a list of values. Click here for an example.

  Example:

  `ciaValue in`

  Operator: in

  Value: 4 5

  matches any incident with a varbind value of either **4** or **5**.

  Note: As shown in the example, each value must be entered on a separate line.

  NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example (**4, 5**). However, the comma-

</td></tr>
</table>

</td></tr>
</table>

Node Settings Suppression Attributes , continued

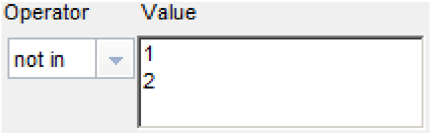
<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td></td><td>

Payload Filter Editor Components, continued

<table>
<tr><th>Attribute</th><th>Description</th></tr>
<tr><td></td><td>
separated list is used only for display purposes. The actual delimiter is the new line.
- **is not null** Finds all non-blank values. Click here for an example.
Example: `ciaValue is not null` matches any incident with a varbind that contains a value.
- **is null** Finds all blank values. Click here for an example.
Example: `ciaValue is null` matches any incident with a varbind that does not contain a value.
- **like** Finds matches using the syntax defined for java regular expressions. See the Pattern (Java Platform SE6) API documentation at : `http://download.oracle.com/javase/6/docs/api/java/util/regex/Pattern.html` for more information. Click here for more information.
The period asterisk (.*) characters mean *any number of characters of any type at this location*.
The period (.) character means *any single character of any type at this location*.
Note: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below.
Example:
`ciaName like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that begins with 1.3.6.1.4.1.9.9 and (optionally) ends with any number of characters.
`ciaValue like .*Chicago.*` finds all traps or events that contain a varbind value that includes the string **Chicago**.
- **not between** Finds all values except those between the two values specified. Click here for an example.
Example: `ciaValue not between 5 8` matches an incident that contains a varbind with the values less than **5** or greater than **8** .
- **not in** Finds all values except those included in the list of values. Click here for an example.
Example:
</td></tr>
</table>

</td></tr>
</table>

Node Settings Suppression Attributes , continued

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td></td><td>
<b>Payload Filter Editor Components, continued</b>
<table>
<tr><th>Attribute</th><th>Description</th></tr>
<tr><td></td><td>
<code>ciaValue not in</code>

Operator
not in
Value
1
2
matches any incident that contains a varbind with values other than <b>1</b> and <b>2</b>.
<b>Note</b>: As shown in the example, each value must be entered on a separate line.
NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example, (<b>1, 2</b>). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line.
• <b>not like</b> Finds all that do not have the values specified using the syntax defined for Java regular expressions. See the Pattern (Java Platform SE6) API documentation at: <code>http://download.oracle.com/javase/6/docs/api/java/util/regex/Pattern.html</code> for more information. Click here for an example.
The period asterisk (.*) characters mean <i>any number of characters of any type at this location</i>.
The period (.) character means <i>any single character of any type at this location</i>.
<b>Note</b>: To include literal string values in the Value attribute, enclose the string value in \Q<<i>literal_value</i>>\E as shown in the Examples listed below.
Example:
<code>ciaName not like \Q.1.3.6.1.4.1.9.9\E.*</code> matches any incident that contains a varbind name value that does not begin with <b>.1.3.6.1.4.1.9.9</b> and (optionally) ends with any number of characters.
<code>ciaValue not like .*Chicago.*</code> finds all traps or events that do not contain a varbind value that includes the string <b>Chicago</b>.
</td></tr>
<tr><td>Value</td><td>The value for which you want NNMi to search.
Note the following:</td></tr>
</table>
</td></tr>
</table>

Node Settings Suppression Attributes , continued

Name	Description

Payload Filter Editor Components, continued

Attribute	Description
	• The values you enter are case sensitive. • NNMi displays a variable number of value fields depending on the Operator selected. For example, the `between` Operator causes two value fields to be displayed. • The `between, in` and `not in` operators require that each value be entered on a separate line.

Payload Filter Editor Buttons

Button	Description
Append	Appends the current expression (Attribute, Operator,and Value) to the selected expression already included in the filter string.
Insert	Inserts the current expression (Attribute, Operator,and Value) in front of the cursor location within the Filter String.
Replace	Replaces the selected expression with the expression displayed in the Attribute, Operator, and Value fields.
AND	Inserts the AND Boolean Operator in the selected cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
OR	Inserts the OR Boolean Operator in the current cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
NOT	Can be used in any part of the Filter String to specify that NNMi should exclude interfaces with values that pass the expression that immediately follows the `NOT`. For example, when evaluating the following expression, NNMi includes interfaces with (interface description) `ifDesc` containing **VLAN**, and excludes any Interfaces that have **VLAN10** for the (interface name) `ifName` value: `(ifDesc like VLAN AND NOT (ifName=VLAN10))` **Note**: View the expression displayed under Filter String to see the logic of the expression as it is created .

Node Settings Suppression Attributes , continued

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td></td><td>
<b>Payload Filter Editor Buttons, continued</b>
<table>
<tr><th>Button</th><th>Description</th></tr>
<tr><td>EXISTS</td><td>Used for filters that include Capabilities or Custom Attribute names and values in the Filer String.
Indicates that you want NNMi to consider interfaces that do not have any Capabilities or Custom Attributes when evaluating the Filter String.
<b>Note</b>: If you include Capabilities or Custom Attribute names and values in the Filer String, but do not use <b>EXISTS</b> or <b>NOT EXISTS</b>, NNMi excludes from its search interfaces that do not include Capabilities or Custom Attributes.
For example, when evaluating the following Filter String, NNMi includes interfaces with (interface description) <code>ifDesc</code> containing <b>VLAN</b>, as well as any Interfaces Custom Attribute Role value is <b>LAN Connection to Oracle Server</b>:
<code>(ifDesc like VLAN OR EXISTS((customAttrName=Role AND customAttrValue=LAN Connection to Oracle Server)))</code>
<b>Note</b>: View the expression displayed under Filter String to see the logic of the expression as it is created.</td></tr>
<tr><td>NOT EXISTS</td><td>Used for filters that include Capabilities or Custom Attribute names and values in the Filer String. Indicates that you want NNMi to consider interfaces that do not have any Capabilities or Custom Attributes when evaluating the Filter String, but exclude the interfaces that match the expression that follows the <b>NOT EXISTS</b>.
<b>Note</b>: If you include Capabilities or Custom Attribute names and values in the Filer String, but do not use <b>EXISTS</b> or <b>NOT EXISTS</b>, NNMi excludes from its search interfaces that do not include Capabilities or Custom Attributes.
For example, when evaluating the following expression, NNMi includes interfaces with (interface description) <code>ifDesc</code> containing <b>VLAN</b>, and excludes any Interfaces that have the Custom Attribute Role and that Role value is <b>LAN Connection to Oracle Server</b>:
<code>(ifDesc like VLAN OR NOT EXISTS ((customAttrName=Role AND customAttrValue=LAN Connection to Oracle Server)))</code>
<b>Note</b>: View the expression displayed under Filter String to see the logic of the expression as it is created.</td></tr>
</table>
</td></tr>
</table>

Node Settings Suppression Attributes , continued

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td></td><td><strong>Payload Filter Editor Buttons, continued</strong>
<table>
<tr><th>Button</th><th>Description</th></tr>
<tr><td>Delete</td><td>Deletes the selected expression.
<strong>Note</strong>: If the Boolean Operator is selected, the Payload Filter Editor deletes all expressions associated with the Boolean Operator.</td></tr>
</table>
</td></tr>
</table>

Configure Incident Enrichment Settings for a Node Group (Remote NNM 6.x/7.x Events)

Note: Node Settings override any other Enrichment settings for this incident, except those configured on the Interface Settings tab.

NNMi enables you to enhanced a specified incident configuration based on the Source Node's participation in a Node Group. The types of items you can fine tune and enhance for a selected incident configuration, include:

- Category
- Family
- Severity
- Priority
- Correlation Nature
- Message
- Assigned To

To configure enrichment settings for a Node Group:

1. Navigate to the **Remote NNM 6.x/7.x Event Configuration** form:
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Expand the **Incidents** folder.
   c. Select **Remote NNM 6.x/7.x Event Configurations** .
   d. Do one of the following:
      i. To create an incident configuration, click the New icon, and continue.
      ii. To edit an incident configuration, select a row, click the Open icon, and continue.
      iii. To delete an incident configuration, select a row and click the Delete icon.
2. Select the **Node Settings** tab.

3. Do one of the following:
   a. To create a new configuration, click the New icon.

   b. To edit an existing configuration, select a row, click the Open icon, and continue.

4. Make sure you configured the basic Node Setting behavior. See "Configure Node Settings for a Remote NNM 6.x/7.x Event Incident" on page 1278for more information.

5. Select the **Enrichment** tab.

6. Do one of the following:

   a. To create an Enrichment configuration, click the New icon and continue.

   b. To edit an Enrichment configuration, select a row, click the Open icon, and continue.

   c. To delete an Enrichment configuration, select a row and click the Delete icon.

7. Configure the desired Enrichment behavior (see the "Enrich Configuration Attributes" table)

8. Click **Save and Close** to save your changes and return to the previous form.

Node Settings Enrich Configuration Attributes

Name	Description
Category	Use the Category attribute to customize the category for this incident configuration. Possible values include: • Accounting • Application Status • Configuration • Fault • Performance • Security • Status See "Specify Category and Family Attribute Values for Organizing Your Incidents (SNMP Trap Incident)" on page 790 for more information.
Family	Use the Family attribute to customize the Family for this incident configuration. Select from the drop-down list or create a new value. For example, some of the values provided by NNMi include: • Address

Node Settings Enrich Configuration Attributes , continued

Name	Description
	• Aggregated Port (Interfaces using **Link Aggregation**[1] protocol. See Interface Form: Link Aggregation tab.) • Card • Connection • Correlation • Interface • Node
Severity	The incident Severity represents the seriousness calculated for the incident. Use the Severity attribute to specify the severity that should be assigned to the incident you are configuring. Possible values are described below: **Normal** - Indicates there are no known problems related to the associated object. This Severity is meant to be informational. Generally, no action is needed for these incidents. **Warning** - Indicates there might be a problem related to the associated object. **Minor** - Indicates NNMi has detected problems related to the associated object that require further investigation. **Major** - Indicates NNMi has detected problems related to the associated object to be resolved before they become critical. **Critical** - Indicates NNMi has detected problems related to the associated object that require immediate attention.
Priority	Used to communicate the urgency of resolving the selected incident. You control this value. NNMi sets this value to null by default. The lower the number the higher the priority. Possible values are: 5 **None** 4 **Low** 3 **Medium** 2 **High**

[1] Protocols used on Switches to configure multiple Interfaces (Aggregation Member Interfaces) to function as if they were one (an Aggregator Interface). When two Aggregator Interfaces establish a connection, that connection is an Aggregator Layer 2 Connection. The Aggregator Layer 2 Connection appears on Layer 2 Neighbor View maps as a thick line with an Interface icon at each end (representing the Aggregator Interface). When you double-click the thick line, it converts into multiple thin lines representing the participating Aggregation Member Layer 2 Connections with their Aggregation Member Interfaces at each end of the lines.

Node Settings Enrich Configuration Attributes , continued

Name	Description
	Top **Note**: The icons are displayed only in table views.
Correlation Nature	Use the Correlation Nature to customize the Correlation Nature for this incident configuration. Possible values include: • Info • None • Root Cause • Secondary Root Cause • Symptom • Stream Correlation • Service Impact • Dedup Stream Correlation • Rate Stream Correlation See Incident Form: General Tab for more information.
Message Format	When configuring an incident, specify how the incident message appears in the incident view. The string you specify in the Message Format attribute is visible in an incident view. **Note**: The incident message limit is 1024 characters. If you exceed this limit, NNMi truncates the value starting from the right. You can use any combination of default and custom attributes: "Valid Parameters for Configuring Incident Messages (SNMP Trap Incident)" on page 795 "Include Custom Incident Attributes in Your Message Format (SNMP Trap Incident)" on page 801
Assigned To	Use to specify the owner of any incident generated for this incident configuration. Click the Lookup icon and select Quick Find to select a valid user name. **Note**: You can also begin to type a valid user name and use the auto-complete feature to select the user name of interest.
Description	Use the Description attribute to provide additional information that you want to note about the current enhancement configuration. This description applies only to the enhancement configuration and does not appear when NNMi displays any associated incident. Type a maximum of 1024 characters. Alpha-numeric and special characters (~ ! @ # $ % ^ & * ( ) _+ -) are permitted.

Configure Custom Incident Attributes to Enrich an Incident Configuration (Node Settings) (Remote NNM 6.x/7.x Events)

The Custom Incident Attributes (CIAs) tab enables you to specify additional CIAs to be saved with an incoming incident. The persisted data might then be used as an argument to an action for the incident.

When creating a CIA for an incident configuration, you can specify any of the following values:

- Custom Attribute on the source node
- Custom Attribute on the interface (source object)

You also specify the Custom Incident Attribute name that will store this information.

For information about each Enrichment tab:

To create a Custom Incident Attribute to enrich an incident configuration:

1. Navigate to the **Remote NNM 6.x/7.x Event Configuration** form:
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Expand the **Incidents** folder.
   c. Select **Remote NNM 6.x/7.x Event Configurations** .
   d. Do one of the following:
      i. To create an incident configuration, click the New icon, and continue.
      ii. To edit an incident configuration, select a row, click the Open icon, and continue.
      iii. To delete an incident configuration, select a row and click the Delete icon.
2. Select the **Node Settings** tab.
3. Do one of the following:
   a. To create a new configuration, click the New icon.
   b. To edit an existing configuration, select a row, click the Open icon, and continue.
   c. To delete an incident configuration, select a row and click the Delete icon.
4. Make sure you configured the basic Node Setting behavior. See "Configure Node Settings for a Remote NNM 6.x/7.x Event Incident" on page 1278 for more information.
5. Select the **Enrichment** tab.
6. Do one of the following:
   a. To create an Enrichment configuration, click the New icon, and continue.
   b. To edit an Enrichment configuration,select a row, click the Open icon, and continue.
   c. To delete an Enrichment configuration, select a row and click the Delete icon.
7. Make sure the Enrichment settings are configure. See "Configure Incident Enrichment Settings for a Node Group (Remote NNM 6.x/7.x Events)" on page 1287 for more information.

8. Navigate to the **Custom Incident Attributes** tab.
9. Do one of the following:
   a. To create a Custom Incident Attribute, click the New icon, and continue.
   b. To edit a Custom Incident Attribute, select a row, click the Open icon, and continue.
   c. To delete a Custom Incident Attribute, select a row and click the Delete icon.
10. Configure the Custom Incident Attribute (see the "Custom Incident Attribute" table).
11. Click **Save and Close** to save your changes and return to the previous form.

Custom Incident Attribute

Name	Description
Custom Incident Attribute Name	Name used to identify the Custom Incident Attribute you are configuring. The name limit is 255 characters. Alpha-numeric characters are permitted. No spaces or special characters (~ ! @ # $ % ^ & * ( ) _+ -) are permitted.
Type	Specifies whether you are using a Custom Attribute on a node or a Custom Attribute on an interface. Possible values are: • Node Custom Attribute • Interface Custom Attribute
Custom Attribute Name	Used to determine the value to be assigned to the Custom Incident Attribute you are configuring. Enter either of the following: • Name of the Custom Attribute on the source node • Name of the Custom Attribute on the interface (source object)

Configure a Payload Filter to Enrich an Incident Configuration (Node Settings) (Remote NNM 6.x/7.x Events)

The Payload Filter Editor enables you to create expressions that further refine the filters used to select the incidents to be enriched. Make sure to design any complex Payload Filters offline as a Boolean expression first. This method can help to minimize errors when entering your expressions using the Payload Filter editor.

To create a Payload Filter expression:

1. Navigate to the **Remote NNM 6.x/7.x Event Configuration** form:
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Expand the **Incidents** folder.
   c. Select **Remote NNM 6.x/7.x Event Configurations**.
   d. Do one of the following:

        i. To create an incident configuration, click the ✳ New icon, and continue.

        ii. To edit an incident configuration, select a row, click the 📂 Open icon, and continue.

        iii. To delete an incident configuration, select a row and click the ✖ Delete icon.

2. Select the **Node Settings** tab.

3. Do one of the following:

    a. To create a new configuration, click the ✳ New icon.

    b. To edit an existing configuration, select a row, click the 📂 Open icon, and continue.

    c. To delete an existing configuration, select a row and click the ✖ Delete icon.

4. Make sure you configured the basic Node Setting behavior. See "Configure Node Settings for a Remote NNM 6.x/7.x Event Incident" on page 1278 for more information.

5. Select the **Enrichment** tab.

6. Do one of the following:

    a. To create an Enrichment configuration, click the ✳ New icon, and continue.

    b. To edit an Enrichment configuration,select a row, click the 📂 Open icon, and continue.

    c. To delete an Enrichment configuration, select a row and click the ✖ Delete icon.

7. Make sure the Enrichment settings are configure. See "Configure Incident Enrichment Settings for a Node Group (Remote NNM 6.x/7.x Events)" on page 1287 for more information.

8. Select the **Payload Filter** tab.

9. Define your Payload Filter (see table). Also see "Guidelines for Creating a Payload Filter".

    a. Plan out the logic needed for your Filter String.

    b. Use the buttons on the bottom half of the Additional Filters Editor to establish the logic structure.

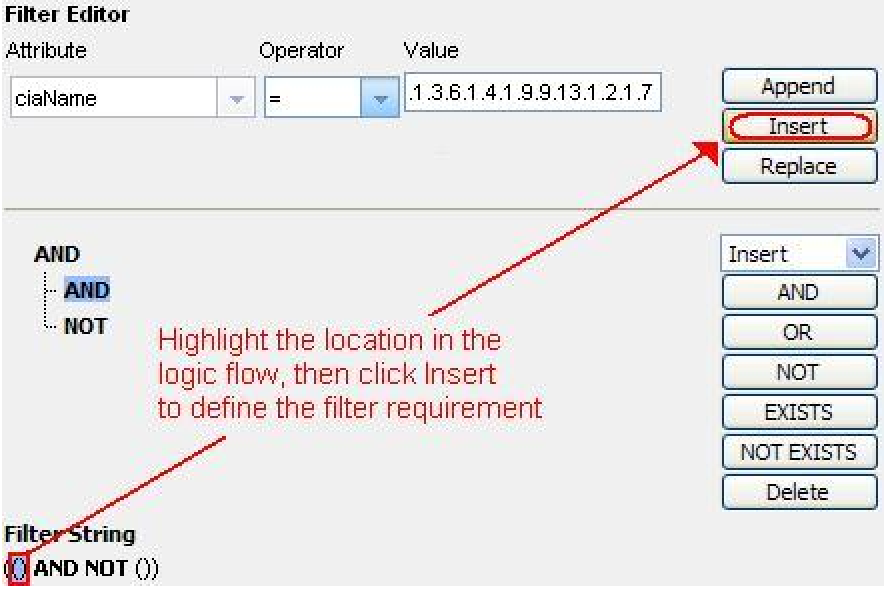
        For example, to establish the following structure, select **Insert**, then click **AND**, then **NOT**, and then **AND** a second time:

        ```
        (( ) AND NOT ( ))
        ```

    c. Now place your cursor in a location within the displayed Filter String, and use the top half of the filter editor to define the parameters of the highlighted filter requirement.

        For example, select a set of parentheses and use the Insert button to specify the filter

requirement within those parentheses:



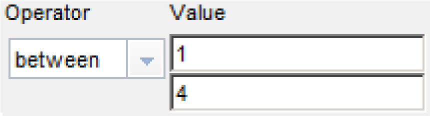
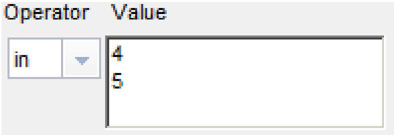
10. Click **Save and Close.**

11. Click **Save and Close** to save your changes and return to the previous form.

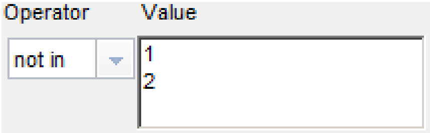
Payload Filter Editor Components

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: • ciaName • ciaValue
Operator	Valid operators are described below. • **=** Finds all values equal to the value specified. Click here for an example. Example: `ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7` matches any incident that contains a varbind with the name value **.1.3.6.1.4.1.9.9.13.1.2.1.7**. • **!=** Finds all values not equal to the value specified. Click here for an example. Example: `ciaName != .1.3.6.1.4.1.9.9.13.1.2.1.7` matches any incident that contains a varbind with the name value other than **1.3.6.1.4.1.9.9.13.1.2.1.7**. • **<** Finds all values less than the value specified. Click here for an example. Example: `ciaValue < 6` matches any incident with a varbind value less than **6**. • **<=** Finds all values less than or equal to the value specified. Click here for an example. Example: `ciaValue <= 6` matches any incident with a varbind value less than or equal to **6**.

Payload Filter Editor Components, continued

Attribute	Description
	• **>** Finds all values greater than the value specified. Click here for an example. Example: `ciaValue > 4` matches any incident with a varbind value greater than **4**. • **>=** Finds all values greater than or equal to the value specified. Click here for an example. Example: `ciaValue >= 4` matches any incident that contains a varbind with values greater than or equal to **4**. • **between** Finds all traps or events that include a varbind with a value equal to and between the two values specified. Click here for an example. Example: `ciaValue between`  Operator: between Value: 1 4 matches any incident that contains a varbind value equal to or greater than **1** and equal to or less than **4**. **Note**: As shown in the example, each value must be entered on a separate line. • **in** Finds any match to at least one value in a list of values. Click here for an example. Example: `ciaValue in`  Operator: in Value: 4 5 matches any incident with a varbind value of either **4** or **5**. **Note**: As shown in the example, each value must be entered on a separate line. NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example (**4, 5**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line. • **is not null** Finds all non-blank values. Click here for an example. Example: `ciaValue is not null` matches any incident with a varbind that contains a value. • **is null** Finds all blank values. Click here for an example. Example: `ciaValue is null` matches any incident with a varbind that does not have a value. • **like** Finds matches using wildcard characters. Click here for more information

Payload Filter Editor Components, continued

Attribute	Description
	about using wildcard characters. The period asterisk (.*) characters mean *any number of characters of any type at this location*. The period (.) character means *any single character of any type at this location*. **Note**: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below. Examples: `ciaName like  \Q.1.3.6.1.4.1.9.9\E.*` finds all traps or events that contain varbind names that begin with **.1.3.6.1.4.1.9.9** and (optionally) end with any number of characters. `ciaValue like .*Chicago.*` finds all traps or events that contain a varbind value that includes the string **Chicago**. • **not between** Finds all values except those between the two values specified. Click here for an example. Example: `ciaValue not between 5 8` matches an incident that contains a varbind with the values less than **5** or greater than **8**. • **not in** Finds all values except those included in the list of values. Click here for an example. Example: `ciaValue not in`  Operator: not in Value: 1 2 matches any incident that contains a varbind with values other than **1** and **2**. **Note**: As shown in the example, each value must be entered on a separate line. NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example, (**1, 2**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line. • **not like** Finds all that do not have the values specified (using wildcard strings). Click here for an example. The period asterisk (.*) characters mean *any number of characters of any type at this location*. The period (.) character means *any single character of any type at this location*. **Note**: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below. Example:

Payload Filter Editor Components, continued

Attribute	Description
	`ciaName not like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that does not begin with **.1.3.6.1.4.1.9.9**. `ciaValue not like .*Chicago.*` finds all traps or events that do not contain a varbind value that includes the string **Chicago**.
Value	The value for which you want NNMi to search. Note the following: • The values you enter are case sensitive. • NNMi displays a variable number of value fields depending on the Operator selected. For example, the `between` Operator causes two value fields to be displayed. • The `between, in` and `not in` operators require that each value be entered on a separate line.

Additional Filters Editor Buttons

Button	Description
Append	Appends the current expression (Attribute, Operator,and Value) to the selected expression already included in the Filter String.
Insert	Inserts the current expression (Attribute, Operator,and Value) in front of the cursor location within the Filter String.
Replace	Replaces the selected expression with the expression displayed in the Attribute, Operator, and Value fields.
AND	Appends, inserts, or replaces the AND Boolean Operator in the selected cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
OR	Appends, Inserts, or replaces the OR Boolean Operator in the current cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
NOT	Can be used in any part of the Filter String to specify that NNMi should exclude interfaces with values that pass the expression that immediately follows the `NOT`. For example, when evaluating the following expression, NNMi includes interfaces with (interface description) `ifDesc` containing **VLAN**, and excludes any Interfaces that have **VLAN10** for the (interface name) `ifName` value: `(ifDesc like VLAN AND NOT (ifName=VLAN10))`

Additional Filters Editor Buttons, continued

Button	Description
	Note: View the expression displayed under Filter String to see the logic of the expression as it is created .
EXISTS	Used for filters that include Capabilities or Custom Attribute names and values in the Filer String. Indicates that you want NNMi to consider interfaces that do not have any Capabilities or Custom Attributes when evaluating the Filter String. **Note**: If you include Capabilities or Custom Attribute names and values in the Filer String, but do not use **EXISTS** or **NOT EXISTS**, NNMi excludes from its search interfaces that do not include Capabilities or Custom Attributes. For example, when evaluating the following Filter String, NNMi includes interfaces with (interface description) `ifDesc` containing **VLAN**, as well as any Interfaces Custom Attribute Role value is **LAN Connection to Oracle Server**: `(ifDesc like VLAN OR EXISTS((customAttrName=Role AND customAttrValue=LAN Connection to Oracle Server)))` **Note**: View the expression displayed under Filter String to see the logic of the expression as it is created.
NOT EXISTS	Used for filters that include Capabilities or Custom Attribute names and values in the Filer String. Indicates that you want NNMi to consider interfaces that do not have any Capabilities or Custom Attributes when evaluating the Filter String, but exclude the interfaces that match the expression that follows the **NOT EXISTS**. Note: If you include Capabilities or Custom Attribute names and values in the Filer String, but do not use **EXISTS** or **NOT EXISTS**, NNMi excludes from its search interfaces that do not include Capabilities or Custom Attributes. For example, when evaluating the following expression, NNMi includes interfaces with (interface description) `ifDesc` containing **VLAN**, and excludes any Interfaces that have the Custom Attribute Role and that Role value is **LAN Connection to Oracle Server**: `(ifDesc like VLAN OR NOT EXISTS((customAttrName=Role AND customAttrValue=LAN Connection to Oracle Server)))` **Note**: View the expression displayed under Filter String to see the logic of the expression as it is created.
Delete	Deletes the selected expression. **Note**: If the Boolean Operator is selected, the Additional Filters Editor deletes all expressions associated with the Boolean Operator.

Configure Incident Dampening Settings for a Node Group (Remote NNM 6.x/7.x Events)

Note: Node Settings override any other Dampening settings for this incident, except those configured on the Interface Settings tab.

NNMi enables you to delay the following for an incident configuration based on the Source Node's participation in a Node Group:

- Execution of Incident Actions
- Execution of Diagnostics (*HP Network Node Manager iSPI Network Engineering Toolset Software \ NNM iSPI NET*)
- Appearance within Incident views in the NNMi Console

Note: You can also suppress the incident configuration based on the Source Object's participation in an Interface Group. See "Configure Incident Dampening Settings for an Interface Group (Remote NNM 6.x/7.x Events)" on page 1262 for more information.

Tip: See "Create Node Groups" on page 295 for more information about Node Groups.

For information about each Node Settings tab:

When using the Dampening configuration, note the following:

- NNMi initially assigns incidents with Dampening settings configured a Lifecycle State of DAMPENED.
- After the dampen interval has passed, NNMi changes the Lifecycle State to REGISTERED.

See About the Incident Lifecycle for more information about Lifecycle State.

To configure Dampening for an incident based on a Node Group:

1. Navigate to the **Remote NNM 6.x/7.x Event Configuration** form:
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Expand **Incidents** folder.
   c. Select **Remote NNM 6.x/7.x Event Configurations** .
   d. Do one of the following:
      i. To create an incident configuration, click the ✱ New icon, and continue.
      ii. To edit an incident configuration, select a row, click the 📂 Open icon, and continue.
      iii. To delete an incident configuration, select a row and click the ✖ Delete icon.
2. Select the **Node Settings** tab.
3. Do one of the following:
   a. To create a new configuration, click the ✱ New icon.
   b. To edit an existing configuration, select a row, click the 📂 Open icon, and continue.

c. To delete an existing configuration, select a row and click the ✖ Delete icon.

4. Make sure you configured the basic Node Setting behavior. See "Configure Node Settings for a Remote NNM 6.x/7.x Event Incident" on page 1278for more information.
5. Select the **Dampen** tab.
6. Configure the desired Dampen behavior (see table).
7. Click **Save and Close** to save your changes and return to the previous form.

Node Settings Dampen Attributes

Name	Description
Enable	Use this attribute to temporarily disable an incident's Dampening settings: **Enable** □ = Temporarily disable the selected configuration. **Enable** ☑ = Enable the selected configuration.
Hour	Specifies the number of hours to be used for the dampen interval.
Minutes	Specifies the number of minutes to be used for the dampen interval. **Note:** The maximum dampen interval is 60 minutes.
Seconds	Specifies the number of seconds to be used for the dampen interval.

Payload Filter	The Payload Filter Editor enables you to create expressions that further refine the filters used to select the incidents to be suppressed, enriched, or dampened. Make sure to design any complex Payload Filters offline as a Boolean expression first. This method can help to minimize errors when entering your expressions using the Payload Filter editor.
	When creating a Payload Filter, note the following:
	• Payload Filter expressions for the `like` and `not like` operators use the syntax defined for java regular expressions (java.util.regex Pattern class).
	• You must use a `ciaName` that already exists in the trap or event you are configuring.
	• Each set of expressions associated with a Boolean Operator is treated as if it were enclosed in parentheses and evaluated together.
	• View the expression displayed under **Filter String** to see the logic of the expression as it is created.
	• The `AND` and `OR` Boolean Operators must contain at least two expressions as shown in the example below.
	The following example filters incidents on voltage state:
	`AND`
	`    ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7`
	`    ciaValue = 5`

Node Settings Dampen Attributes , continued

Name	Description

	NNMi evaluates the expression above as follows:
	`(ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7 AND ciaValue = 5)`
	NNMi finds all incidents with a varbind `.1.3.6.1.4.1.9.9.13.1.2.1.7` value of **5**.
	• The placement of your cursor and the subsequent text that is selected is important when performing operations using the Payload Filter Editor. For example, you append to, replace, or change the indentation of the expression that is selected.
	• The placement of your cursor and the subsequent text that is selected is especially important when adding your Boolean operators.
	• You can include more than one varbind in the same Payload Filter expression as shown in the following example:
	`((ciaName like \Q.1.3.6.1.4.1.9.9\E.* AND ciaValue = 25) AND (ciaName like \Q.1.3.6.1.2.1.2.2.1.1.3\E.* AND ciaValue = 3))`
	In this example, a given trap must meet each of the following criteria:
	▪ Contain a varbind whose Object Identifier (OID) matches the regular expression `\Q.1.3.6.1.4.1.9.9\E.*` and has a value of `25`.
	▪ Contain a varbind whose OID matches the regular expression `\Q.1.3.6.1.2.1.2.2.1.1.3\E.*` and has a value of `3`.

Payload Filter Editor Components

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: • ciaName • ciaValue
Operator	Valid operators are described below. • **=** Finds all values equal to the value specified. Click here for an example. Example: `ciaName=.1.3.6.1.4.1.9.9.13.1.2.1.7` matches any incident that contains a varbind with the name value of **.1.3.6.1.4.1.9.9.13.1.2.1.7**. • **!=** Finds all values not equal to the value specified. Click here for an example. Example: `ciaName! = .1.3.6.1.4.1.9.9.13.1.2.1.7` matches any incident that contains a varbind with a name value other than

Node Settings Dampen Attributes , continued

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td></td><td>

Payload Filter Editor Components, continued

<table>
<tr><th>Attribute</th><th>Description</th></tr>
<tr><td></td><td>

1.3.6.1.4.1.9.9.13.1.2.1.7.

- **<** Finds all values less than the value specified. Click here for an example.

  Example: `ciaValue < 6` matches any incident that contains a varbind with a value less than **6**.

- **<=** Finds all values less than or equal to the value specified. Click here for an example.

  Example: `ciaValue <= 6` matches any incident that contains a varbind with a value less than or equal to **6**.

- **>** Finds all values greater than the value specified. Click here for an example.

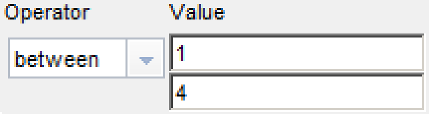
  Example: `ciaValue > 4` matches any incident that contains a varbind with a value greater than **4**.

- **>=** Finds all values greater than or equal to the value specified. Click here for an example.

  Example: `ciaValue >= 4` matches any incident that contains a varbind with values greater than or equal to **4**.

- **between** Finds all values equal to and between the two values specified. Click here for an example.

  Example: `ciaValue between`

  Operator: between
  Value: 1
  4

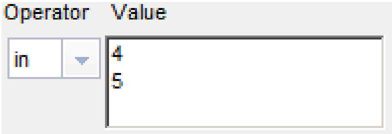
  matches any incident that contains a varbind value equal to or greater than **1** and equal to or less than **4**.

  Note: As shown in the example, each value must be entered on a separate line.

- **in** Finds any match to at least one value in a list of values. Click here for an example.

  Example:

  `ciaValue in`

  Operator: in
  Value: 4
  5

</td></tr>
</table>

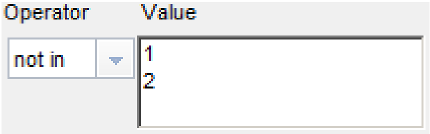
</td></tr>
</table>

Node Settings Dampen Attributes , continued

Name	Description
	Payload Filter Editor Components, continued

Attribute	Description
	matches any incident with a varbind value of either **4** or **5**. **Note**: As shown in the example, each value must be entered on a separate line. NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example (**4, 5**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line. - **is not null** Finds all non-blank values. Click here for an example. Example: `ciaValue is not null` matches any incident with a varbind that contains a value. - **is null** Finds all blank values. Click here for an example. Example: `ciaValue is null` matches any incident with a varbind that does not contain a value. - **like** Finds matches using the syntax defined for java regular expressions. See the Pattern (Java Platform SE6) API documentation at : `http://download.oracle.com/javase/6/docs/api/java/util/regex/Pattern.html` for more information. Click here for more information. The period asterisk (.*) characters mean *any number of characters of any type at this location*. The period (.) character means *any single character of any type at this location*. **Note**: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below. Example: `ciaName like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that begins with 1.3.6.1.4.1.9.9 and (optionally) ends with any number of characters. `ciaValue like .*Chicago.*` finds all traps or events that contain a varbind value that includes the string **Chicago**. - **not between** Finds all values except those between the two values specified. Click here for an example. Example: `ciaValue not between 5 8` matches an incident that

Node Settings Dampen Attributes , continued

Name	Description
	Payload Filter Editor Components, continued <table><tr><th>Attribute</th><th>Description</th></tr><tr><td></td><td>contains a varbind with the values less than **5** or greater than **8** . - **not in** Finds all values except those included in the list of values. Click here for an example. Example: `ciaValue not in`  Operator: not in Value: 1 2 matches any incident that contains a varbind with values other than **1** and **2**. **Note**: As shown in the example, each value must be entered on a separate line. NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example, (**1, 2**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line. - **not like** Finds all that do not have the values specified using the syntax defined for Java regular expressions. See the Pattern (Java Platform SE6) API documentation at: `http://download.oracle.com/javase/6/docs/api/java/util/regex/Pattern.html` for more information. Click here for an example. The period asterisk (.*) characters mean *any number of characters of any type at this location*. The period (.) character means *any single character of any type at this location*. **Note**: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below. Example: `ciaName not like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that does not begin with **.1.3.6.1.4.1.9.9** and (optionally) ends with any number of characters.</td></tr></table>

Node Settings Dampen Attributes , continued

Name	Description

Payload Filter Editor Components, continued

Attribute	Description
	`ciaValue not like .*Chicago.*` finds all traps or events that do not contain a varbind value that includes the string **Chicago**.
Value	The value for which you want NNMi to search. Note the following: • The values you enter are case sensitive. • NNMi displays a variable number of value fields depending on the Operator selected. For example, the `between` Operator causes two value fields to be displayed. • The `between, in` and `not in` operators require that each value be entered on a separate line.

Payload Filter Editor Buttons

Button	Description
Append	Appends the current expression (Attribute, Operator,and Value) to the selected expression already included in the filter string.
Insert	Inserts the current expression (Attribute, Operator,and Value) in front of the cursor location within the Filter String.
Replace	Replaces the selected expression with the expression displayed in the Attribute, Operator, and Value fields.
AND	Inserts the AND Boolean Operator in the selected cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
OR	Inserts the OR Boolean Operator in the current cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
NOT	Can be used in any part of the Filter String to specify that NNMi should exclude interfaces with values that pass the expression that immediately follows the `NOT`. For example, when evaluating the following expression, NNMi includes interfaces with (interface description) `ifDesc` containing **VLAN**, and excludes any Interfaces that have **VLAN10** for the (interface name) `ifName` value:

Node Settings Dampen Attributes , continued

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td></td><td>

Payload Filter Editor Buttons, continued

<table>
<tr><th>Button</th><th>Description</th></tr>
<tr><td></td><td><code>(ifDesc like VLAN AND NOT (ifName=VLAN10))</code>
<b>Note</b>: View the expression displayed under Filter String to see the logic of the expression as it is created .</td></tr>
<tr><td>EXISTS</td><td>Used for filters that include Capabilities or Custom Attribute names and values in the Filer String.
Indicates that you want NNMi to consider interfaces that do not have any Capabilities or Custom Attributes when evaluating the Filter String.
<b>Note</b>: If you include Capabilities or Custom Attribute names and values in the Filer String, but do not use <b>EXISTS</b> or <b>NOT EXISTS</b>, NNMi excludes from its search interfaces that do not include Capabilities or Custom Attributes.
For example, when evaluating the following Filter String, NNMi includes interfaces with (interface description) <code>ifDesc</code> containing <b>VLAN</b>, as well as any Interfaces Custom Attribute Role value is <b>LAN Connection to Oracle Server</b>:
<code>(ifDesc like VLAN OR EXISTS((customAttrName=Role AND customAttrValue=LAN Connection to Oracle Server)))</code>
<b>Note</b>: View the expression displayed under Filter String to see the logic of the expression as it is created.</td></tr>
<tr><td>NOT EXISTS</td><td>Used for filters that include Capabilities or Custom Attribute names and values in the Filer String. Indicates that you want NNMi to consider interfaces that do not have any Capabilities or Custom Attributes when evaluating the Filter String, but exclude the interfaces that match the expression that follows the <b>NOT EXISTS</b>.
<b>Note</b>: If you include Capabilities or Custom Attribute names and values in the Filer String, but do not use <b>EXISTS</b> or <b>NOT EXISTS</b>, NNMi excludes from its search interfaces that do not include Capabilities or Custom Attributes.
For example, when evaluating the following expression, NNMi includes interfaces with (interface description) <code>ifDesc</code> containing <b>VLAN</b>, and excludes any Interfaces that have the Custom Attribute Role and that Role value is <b>LAN Connection to Oracle Server</b>:
<code>(ifDesc like VLAN OR NOT EXISTS ((customAttrName=Role AND customAttrValue=LAN</code></td></tr>
</table>

</td></tr>
</table>

Node Settings Dampen Attributes , continued

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td></td><td>
<b>Payload Filter Editor Buttons, continued</b>
<table>
<tr><th>Button</th><th>Description</th></tr>
<tr><td></td><td><code>Connection to Oracle Server)))</code>
<b>Note</b>: View the expression displayed under Filter String to see the logic of the expression as it is created.</td></tr>
<tr><td>Delete</td><td>Deletes the selected expression.
<b>Note</b>: If the Boolean Operator is selected, the Payload Filter Editor deletes all expressions associated with the Boolean Operator.</td></tr>
</table>
</td></tr>
</table>

Configure Incident Actions for a Node Group (Remote NNM 6.x/7.x Events)

Note: Interface Settings override any other Actions settings for this incident, including those from the Node Settings tab.

For information about each Node Settings tab:

NNMi enables you to configure incident actions based on a Source Node's participation in a Node Group.

You can configure actions to automatically run at any point in the incident lifecycle. For example, you might want to configure an action to occur when an incident of the type you are configuring is generated (**Registered**). When an incident is generated, you might want to automatically open a trouble ticket or send email or page your network operator. After the incident is **Closed**, you might want to automatically close the trouble ticket.

Note: Your actions will not be executed until you enable the Actions configuration by either clicking Enable ☑ on the Actions tab or using the **Actions** → **Enable Configuration** option.

You can configure actions for incidents generated from SNMP traps, NNM 6.x or 7.x events, and the NNMi management events. Any time an incident configuration changes, the action directory is rescanned and any Jython files are reloaded to the NNMi database. See "Lifecycle Transition Action Form (Remote NNM 6.x/7.x Events)" on page 1358 for more information about the actions directory.

Tip: Copy any required Jython files to the NNMi actions directory before you configure an incident action. New or updated actions are loaded into NNMi only when an incident configuration is updated or created. See "Lifecycle Transition Action Form (Remote NNM 6.x/7.x Events)" on page 1358 for the location of the NNMi action directory.

When the action runs, output is logged to the `incidentActions.*.*.log` file. See "Verify that NNMi Services are Running" on page 85 for more information about log files and where they are located.

To configure an automatic action for an incident:

1. Navigate to the **Remote NNM 6.x/7.x Event Configuration** form.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Expand the **Incidents** folder.
   c. Select **Remote NNM 6.x/7.x Event Configurations**.
   d. Do one of the following:
      i. To create a new incident configuration, click the New icon.
      ii. To edit an existing incident configuration, select a row, click the Open icon, and continue.
2. Select the **Node Settings** tab.
3. Do one of the following:
   a. To create a new configuration, click the New icon.
   b. To edit an existing configuration, select a row, click the Open icon, and continue.
   c. To delete an existing configuration, select a row and click the Delete icon.
4. Make sure you configured the basic Node Setting behavior. See "Configure Node Settings for a Remote NNM 6.x/7.x Event Incident" on page 1278 for more information.
5. Select the **Actions** tab.
6. From the **Lifecycle Actions** table toolbar, do one of the following:
   - To create an Action configuration, click the New icon, and continue.
   - To edit an Action configuration, select a row, click the Open icon, and continue.
   - To delete an Action configuration, select a row and click the Delete icon.
7. In the "Lifecycle Transition Action Form (Remote NNM 6.x/7.x Events)" on page 1358, provide the required information.
8. Click **Save and Close** to save your changes and return to the previous form.

   The next time the lifecycle changes, NNMi launches the action associated with the lifecycle for the incident of that type.

Configure a Payload Filter for an Incident Action (Node Settings) (Remote NNM 6.x/7.x Events)

The Payload Filter Editor enables you to create expressions that further refine the filters used to select the incidents that cause the configured action to run. Make sure to design any complex Payload Filters offline as a Boolean expression first. This method can help to minimize errors when entering your expressions using the Payload Filter editor.

To create a Payload Filter expression:

1. Navigate to the **Remote NNM 6.x/7.x Event Configuration** form:
   a. From the workspace navigation panel, select the **Configuration** workspace.

   b. Expand the **Incidents** folder.

   c. Select **Remote NNM 6.x/7.x Event Configurations**.

   d. Do one of the following:

      i. To create an incident configuration, click the ✳ New icon, and continue.

      ii. To edit an incident configuration, select a row, click the 📂 Open icon, and continue.

      iii. To delete an incident configuration, select a row and click the ✖ Delete icon.

2. Select the **Node Settings** tab.

3. Do one of the following:

   - To create a new configuration, click the ✳ New icon.

   - To edit a configuration, select a row, click the 📂 Open icon, and continue.

4. Make sure you configure the basic Node Setting behavior. See "Configure Node Settings for a Remote NNM 6.x/7.x Event Incident" on page 1278 for more information.

5. Select the **Actions** tab.

6. Do one of the following:

   - To create an Action configuration, click the ✳ New icon, and continue.

   - To edit an Action configuration, select a row, click the 📂 Open icon, and continue.

   - To delete an Action configuration, select a row and click the ✖ Delete icon.

7. Make sure you configured the Action Configuration settings. See "Configure Incident Actions for a Node Group (Remote NNM 6.x/7.x Events)" on page 1307 for more information.

8. Select the **Payload Filter** tab.

9. Define your Payload Filter (see table). Also see "Guidelines for Creating a Payload Filter".

   a. Plan out the logic needed for your Filter String.

   b. Use the buttons on the bottom half of the Additional Filters Editor to establish the logic structure.

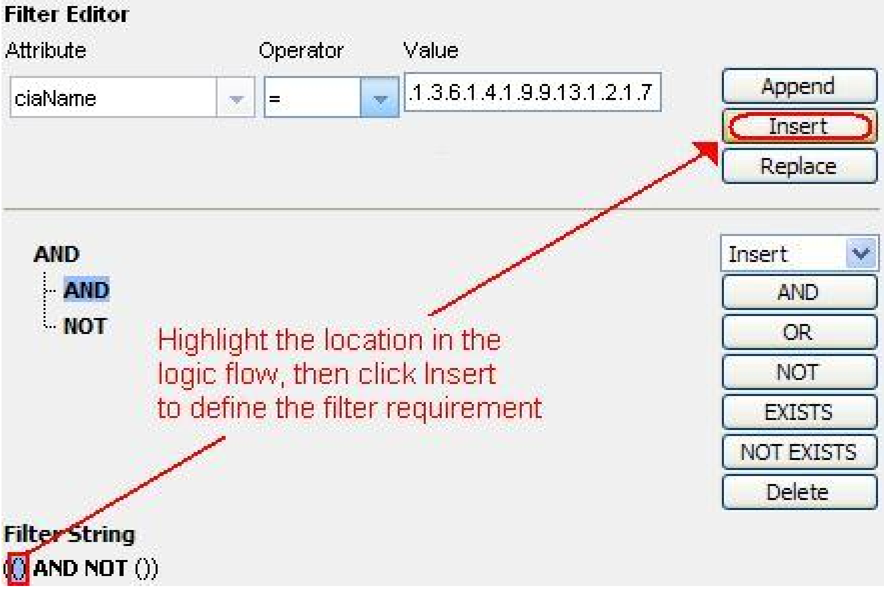
      For example, to establish the following structure, select **Insert**, then click **AND**, then **NOT**, and then **AND** a second time:

      ```
      (( ) AND NOT ( ))
      ```

   c. Now place your cursor in a location within the displayed Filter String, and use the top half of the filter editor to define the parameters of the highlighted filter requirement.

      For example, select a set of parentheses and use the Insert button to specify the filter

requirement within those parentheses:



Highlight the location in the logic flow, then click Insert to define the filter requirement

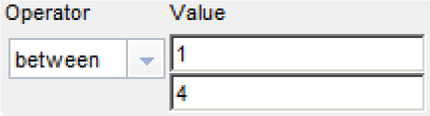
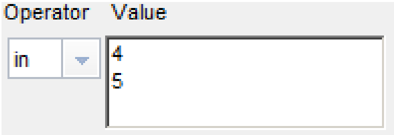
10. Click Save and Close.

11. Click Save and Close to save your changes and return to the previous form.

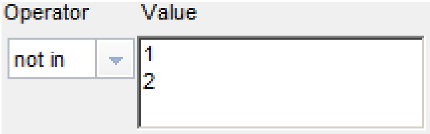
Payload Filter Editor Components

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: • ciaName • ciaValue
Operator	Valid operators are described below. • **=** Finds all values equal to the value specified. Click here for an example. Example: `ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7` matches any incident that contains a varbind with the name value **.1.3.6.1.4.1.9.9.13.1.2.1.7**. • **!=** Finds all values not equal to the value specified. Click here for an example. Example: `ciaName != .1.3.6.1.4.1.9.9.13.1.2.1.7` matches any incident that contains a varbind with the name value other than **1.3.6.1.4.1.9.9.13.1.2.1.7**. • **<** Finds all values less than the value specified. Click here for an example. Example: `ciaValue < 6` matches any incident with a varbind value less than **6**. • **<=** Finds all values less than or equal to the value specified. Click here for an example. Example: `ciaValue <= 6` matches any incident with a varbind value less than or equal to **6**.

Payload Filter Editor Components, continued

Attribute	Description
	- **>** Finds all values greater than the value specified. Click here for an example. Example: `ciaValue > 4` matches any incident with a varbind value greater than **4**. - **>=** Finds all values greater than or equal to the value specified. Click here for an example. Example: `ciaValue >= 4` matches any incident that contains a varbind with values greater than or equal to **4**. - **between** Finds all traps or events that include a varbind with a value equal to and between the two values specified. Click here for an example. Example: `ciaValue between`  Operator: between Value: 1 4 matches any incident that contains a varbind value equal to or greater than **1** and equal to or less than **4**. **Note**: As shown in the example, each value must be entered on a separate line. - **in** Finds any match to at least one value in a list of values. Click here for an example. Example: `ciaValue in`  Operator: in Value: 4 5 matches any incident with a varbind value of either **4** or **5**. **Note**: As shown in the example, each value must be entered on a separate line. NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example (**4, 5**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line. - **is not null** Finds all non-blank values. Click here for an example. Example: `ciaValue is not null` matches any incident with a varbind that contains a value. - **is null** Finds all blank values. Click here for an example. Example: `ciaValue is null` matches any incident with a varbind that does not have a value. - **like** Finds matches using wildcard characters. Click here for more information

Payload Filter Editor Components, continued

Attribute	Description
	about using wildcard characters. The period asterisk (.*) characters mean *any number of characters of any type at this location*. The period (.) character means *any single character of any type at this location*. **Note**: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below. Examples: `ciaName like  \Q.1.3.6.1.4.1.9.9\E.*` finds all traps or events that contain varbind names that begin with **.1.3.6.1.4.1.9.9** and (optionally) end with any number of characters. `ciaValue like .*Chicago.*` finds all traps or events that contain a varbind value that includes the string **Chicago**. • **not between** Finds all values except those between the two values specified. Click here for an example. Example: `ciaValue not between 5 8` matches an incident that contains a varbind with the values less than **5** or greater than **8**. • **not in** Finds all values except those included in the list of values. Click here for an example. Example: `ciaValue not in`  Operator: not in Value: 1 2 matches any incident that contains a varbind with values other than **1** and **2**. **Note**: As shown in the example, each value must be entered on a separate line. NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example, (**1, 2**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line. • **not like** Finds all that do not have the values specified (using wildcard strings). Click here for an example. The period asterisk (.*) characters mean *any number of characters of any type at this location*. The period (.) character means *any single character of any type at this location*. **Note**: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below. Example:

Payload Filter Editor Components, continued

Attribute	Description
	`ciaName not like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that does not begin with **.1.3.6.1.4.1.9.9**. `ciaValue not like .*Chicago.*` finds all traps or events that do not contain a varbind value that includes the string **Chicago**.
Value	The value for which you want NNMi to search. Note the following: • The values you enter are case sensitive. • NNMi displays a variable number of value fields depending on the Operator selected. For example, the `between` Operator causes two value fields to be displayed. • The `between, in` and `not in` operators require that each value be entered on a separate line.

Additional Filters Editor Buttons

Button	Description
Append	Appends the current expression (Attribute, Operator,and Value) to the selected expression already included in the Filter String.
Insert	Inserts the current expression (Attribute, Operator,and Value) in front of the cursor location within the Filter String.
Replace	Replaces the selected expression with the expression displayed in the Attribute, Operator, and Value fields.
AND	Appends, inserts, or replaces the AND Boolean Operator in the selected cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
OR	Appends, Inserts, or replaces the OR Boolean Operator in the current cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
NOT	Can be used in any part of the Filter String to specify that NNMi should exclude interfaces with values that pass the expression that immediately follows the `NOT`. For example, when evaluating the following expression, NNMi includes interfaces with (interface description) `ifDesc` containing **VLAN**, and excludes any Interfaces that have **VLAN10** for the (interface name) `ifName` value: `(ifDesc like VLAN AND NOT (ifName=VLAN10))`

Additional Filters Editor Buttons, continued

Button	Description
	Note: View the expression displayed under Filter String to see the logic of the expression as it is created .
EXISTS	Used for filters that include Capabilities or Custom Attribute names and values in the Filer String. Indicates that you want NNMi to consider interfaces that do not have any Capabilities or Custom Attributes when evaluating the Filter String. **Note**: If you include Capabilities or Custom Attribute names and values in the Filer String, but do not use **EXISTS** or **NOT EXISTS**, NNMi excludes from its search interfaces that do not include Capabilities or Custom Attributes. For example, when evaluating the following Filter String, NNMi includes interfaces with (interface description) `ifDesc` containing **VLAN**, as well as any Interfaces Custom Attribute Role value is **LAN Connection to Oracle Server**: `(ifDesc like VLAN OR EXISTS((customAttrName=Role AND customAttrValue=LAN Connection to Oracle Server)))` **Note**: View the expression displayed under Filter String to see the logic of the expression as it is created.
NOT EXISTS	Used for filters that include Capabilities or Custom Attribute names and values in the Filer String. Indicates that you want NNMi to consider interfaces that do not have any Capabilities or Custom Attributes when evaluating the Filter String, but exclude the interfaces that match the expression that follows the **NOT EXISTS**. Note: If you include Capabilities or Custom Attribute names and values in the Filer String, but do not use **EXISTS** or **NOT EXISTS**, NNMi excludes from its search interfaces that do not include Capabilities or Custom Attributes. For example, when evaluating the following expression, NNMi includes interfaces with (interface description) `ifDesc` containing **VLAN**, and excludes any Interfaces that have the Custom Attribute Role and that Role value is **LAN Connection to Oracle Server**: `(ifDesc like VLAN OR NOT EXISTS((customAttrName=Role AND customAttrValue=LAN Connection to Oracle Server)))` **Note**: View the expression displayed under Filter String to see the logic of the expression as it is created.
Delete	Deletes the selected expression. **Note**: If the Boolean Operator is selected, the Additional Filters Editor deletes all expressions associated with the Boolean Operator.

Configure Diagnostics Selections for a Node Group (Remote NNM 6.x/7.x Events)

Note: Node Settings override any other Diagnostics Selections settings for this incident, except those configured on the Interface Settings tab.

For information about each Node Settings tab: .

(*HP Network Node Manager iSPI Network Engineering Toolset Software*) The Diagnostic Selections form enables you to configure NNMi to automatically gather NNM iSPI NET diagnostic information for the Incident you are configuring. When using this form, you specify the diagnostics you want to run on each applicable node in the specified Node Group.

To configure Diagnostics to run on a Source Node for an incident:

1. Navigate to the **Diagnostics Selection** form.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Expand the **Incidents** folder.
   c. Select **Remote NNM 6.x/7.x Event Configurations** .
   d. Do one of the following:
      - To create an incident configuration, click the New icon.
      - To edit an incident configuration, select a row, click the Open icon, and continue.
   e. Navigate to **Node Settings** tab, and do one of the following:
      - To create a Node Settings configuration, click the New icon, and continue.
      - To edit a Node Settings configuration, select a row, click the Open icon, and continue.
      - To delete a Node Settings configuration, select the Node setting, and click the Delete icon.
   f. Navigate to the **Diagnostic Selection** tab, and do one of the following:
      - To create a Diagnostic Selection setting, click the New icon, and continue.
      - To edit a Diagnostic Selection setting, select a row, click the Open icon, and continue.
      - To delete a Diagnostic Selection setting, select the Diagnostic Selection setting, and click the Delete icon.
2. Provide the required information (see table).
3. Click **Save and Close** to save your changes and return to the previous form.

After you configure the Diagnostic for the incident and Node Group, the Diagnostic must match the following criteria before the Diagnostic runs:

- The Source Node must be in the specified Node Group.
- The Diagnostic must be valid for the Source Node. (For example, only Nortel switch Diagnostics are run on Nortel switches.)
- The incident’s current lifecycle state must match a lifecycle state for which it was configured. (For example, if you configure the Incident to run a specified Diagnostic when the incident is Closed, then if the current Incident's Lifecycle State is Closed, NNMi runs that Diagnostic.)

Note: If a Source Node is in more than one Node Group, the Diagnostic is only run on the node the first time NNMi finds a match for that Source Node based on the configuration Ordering field.

If these criteria are met, NNM iSPI NET runs the Diagnostics and generates Diagnostic reports to help you solve the problem on the Source Node.

After you configure Diagnostics for an incident, you can also run Diagnostics and access the Diagnostics reports on demand, using **Actions** → **Run Diagnostics (iSPI Net only)** in the Incident form. The same criteria apply (see the criteria above). See Incident Form:Diagnostics Tab for more information.

Tip: You can right-click any object in a table or map view to access the **Actions** menu.

You can also run and access Diagnostics reports from a Node form. See Node Form: Diagnostics Tab for more information.

Diagnostic Settings Attributes

Attribute	Description
Flow Definition	Select the Diagnostic (Flow Definition) you want to use for the specified Node Group. Click the Lookup icon and choose one of the following options: • Show Analysis to display Analysis Pane information for the current Diagnostic (Flow Definition). (See Use the Analysis Pane for more information about the Analysis Pane.) • Quick Find to view the list of possible diagnostic Flow Definitions. NNMi provides diagnostics for the following types of devices: ▪ Cisco switch ▪ Cisco router ▪ Cisco switch/router ▪ Nortel switch See "Diagnostics (Flows) Provided by NNM iSPI NET" on page 758 for more information about the diagnostics provided and the devices to which they apply.
Lifecycle State	Incident Lifecycle State of the target Incident. If the incident's Lifecycle State matches the value specified here, the Diagnostic runs. The Diagnostic automatically runs on each applicable Source Node in the specified

Diagnostic Settings Attributes, continued

Attribute	Description
	Node Group if the incident has the Lifecycle State currently configured in this attribute of the Diagnostic (Flow Definition - set of automated commands).
Enable	Use this attribute to temporarily disable an incident's Diagnostics settings. To temporarily disable the selected Diagnostics settings, clear **Enable**□. To enable the selected Diagnostics settings, click **Enable**☑.

Configure Suppression Settings for a Remote NNM 6.x/7.x Event Incident

For information about each Remote NNM 6.x/7.x Events tab:

NNMi enables you to suppress incidents based on Interface Group, Node Group, or default Suppression settings. NNMi applies your Suppression settings in the following order. Only the first match applies.

1. Interface Group (Remote 6.x/7.x Event Configuration Form: Interface Settings tab)
2. Node Group (Remote 6.x/7.x Event Configuration Form: Node Settings tab)
3. Enrich configuration settings without specifying an Interface Group or Node Group (Remote 6.x/7.x Event Configuration Form: Enrichment tab)

A Payload Filter enables you to use the data that is included with any of the following items before they are stored as incidents in NNMi:

- Traps generated from an SNMP agent
- Management incidents that are generated by NNMi
- Events generated by NNM 6.x or 7.x management stations

Examples of the type of data that can be used as a Payload Filter include SNMP trap varbind names and values as well as CIA (Custom Incident Attribute) names and values. For example, you might want NNMi to suppress a particular status change notification trap for a specified Node Group or Interface Group. To do so, you could include the name of the trap varbind that stores this information as well as the particular status change value string the traps that you want to suppress should contain.

See "Configure Incident Suppression Settings for an Interface Group (Remote NNM 6.x/7.x Events)" on page 1242 for information about how to suppress an incident for an Interface Group with or without a Payload Filter.

See "Configure Incident Suppression Settings for a Node Group (Remote NNM 6.x/7.x Events)" on page 1279 for more information about how to suppress an incident for a Node Group with or without a Payload Filter.

To configure suppression for an incident using a Payload Filter without an Interface Group or Node Group Filter:

1. Navigate to the **Remote NNM 6.x/7.x Event Configuration** form:
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Expand the **Incidents** folder.
   c. Select **Remote NNM 6.x/7.x Event Configurations**.
   d. Do one of the following:
      i. To create an incident configuration, click the New icon, and continue.
      ii. To edit an incident configuration, select a row, click the Open icon, and continue.
      iii. To delete an incident configuration, select a row and click the Delete icon.
2. Select the **Suppression** tab.
3. Provide the required information (see table)
4. Click **Save and Close** to save your configuration return to the previous form.

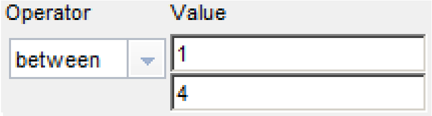
Suppression Configuration Attributes

Name	Description
Enable	Use this attribute to temporarily disable an incident's suppression settings. To temporarily disable the Suppression settings for the selected incident configuration, clear **Enable** ☐. To enable the Suppression settings for the selected incident configuration, click **Enable** ☑.
Payload Filter	The Payload Filter Editor enables you to create expressions that further refine the filters used to select the incidents to be suppressed, enriched, or dampened. Make sure to design any complex Payload Filters offline as a Boolean expression first. This method can help to minimize errors when entering your expressions using the Payload Filter editor. When creating a Payload Filter, note the following: • Payload Filter expressions for the `like` and `not like` operators use the syntax defined for java regular expressions (java.util.regex Pattern class). • You must use a `ciaName` that already exists in the trap or event you are configuring. • Each set of expressions associated with a Boolean Operator is treated as if it were enclosed in parentheses and evaluated together. • View the expression displayed under **Filter String** to see the logic of the expression as it is created. • The `AND` and `OR` Boolean Operators must contain at least two expressions as shown in the example below. The following example filters incidents on voltage state: `AND`

Suppression Configuration Attributes , continued

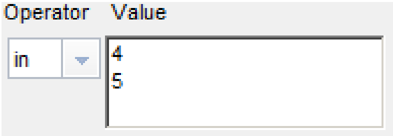
<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td></td><td>
<pre>
    ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7
    ciaValue = 5
</pre>
<p>NNMi evaluates the expression above as follows:</p>
<pre>(ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7 AND ciaValue = 5)</pre>
<p>NNMi finds all incidents with a varbind .1.3.6.1.4.1.9.9.13.1.2.1.7 value of <b>5</b>.</p>
<ul>
<li>The placement of your cursor and the subsequent text that is selected is important when performing operations using the Payload Filter Editor. For example, you append to, replace, or change the indentation of the expression that is selected.</li>
<li>The placement of your cursor and the subsequent text that is selected is especially important when adding your Boolean operators.</li>
<li>You can include more than one varbind in the same Payload Filter expression as shown in the following example:
<pre>((ciaName like \Q.1.3.6.1.4.1.9.9\E.* AND ciaValue = 25) AND
(ciaName like \Q.1.3.6.1.2.1.2.2.1.1.3\E.* AND ciaValue = 3))</pre>
<p>In this example, a given trap must meet each of the following criteria:</p>
<ul>
<li>Contain a varbind whose Object Identifier (OID) matches the regular expression \Q.1.3.6.1.4.1.9.9\E.* and has a value of 25.</li>
<li>Contain a varbind whose OID matches the regular expression \Q.1.3.6.1.2.1.2.2.1.1.3\E.* and has a value of 3.</li>
</ul>
</li>
</ul>
<p><b>Payload Filter Editor Components</b></p>
<table>
<tr><th>Attribute</th><th>Description</th></tr>
<tr><td>Attribute</td><td>The attribute name on which NNMi searches. Filterable attributes include the following:
<ul>
<li>ciaName</li>
<li>ciaValue</li>
</ul></td></tr>
<tr><td>Operator</td><td>Valid operators are described below.
<ul>
<li><b>=</b> Finds all values equal to the value specified. Click here for an example.
<p>Example: ciaName=.1.3.6.1.4.1.9.9.13.1.2.1.7 matches any incident that contains a varbind with the name value of <b>.1.3.6.1.4.1.9.9.13.1.2.1.7</b>.</p></li>
<li><b>!=</b> Finds all values not equal to the value specified. Click here for an example.</li>
</ul></td></tr>
</table>
</td></tr>
</table>

Suppression Configuration Attributes , continued

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td></td><td>
<b>Payload Filter Editor Components, continued</b>
<table>
<tr><th>Attribute</th><th>Description</th></tr>
<tr><td></td><td>
Example: `ciaName! = .1.3.6.1.4.1.9.9.13.1.2.1.7` matches any incident that contains a varbind with a name value other than <b>1.3.6.1.4.1.9.9.13.1.2.1.7</b>.
- <b>&lt;</b> Finds all values less than the value specified. Click here for an example.
Example: `ciaValue < 6` matches any incident that contains a varbind with a value less than <b>6</b>.
- <b>&lt;=</b> Finds all values less than or equal to the value specified. Click here for an example.
Example: `ciaValue <= 6` matches any incident that contains a varbind with a value less than or equal to <b>6</b>.
- <b>&gt;</b> Finds all values greater than the value specified. Click here for an example.
Example: `ciaValue > 4` matches any incident that contains a varbind with a value greater than <b>4</b>.
- <b>&gt;=</b> Finds all values greater than or equal to the value specified. Click here for an example.
Example: `ciaValue >= 4` matches any incident that contains a varbind with values greater than or equal to <b>4</b>.
- <b>between</b> Finds all values equal to and between the two values specified. Click here for an example.
Example: `ciaValue between`

Operator: between
Value: 1
4
matches any incident that contains a varbind value equal to or greater than <b>1</b> and equal to or less than <b>4</b>.
<b>Note</b>: As shown in the example, each value must be entered on a separate line.
- <b>in</b> Finds any match to at least one value in a list of values. Click here for an example.
Example:
`ciaValue in`
</td></tr>
</table>
</td></tr>
</table>

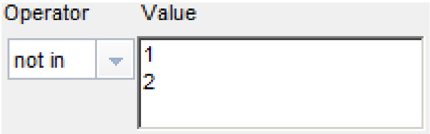
Suppression Configuration Attributes , continued

Name	Description
	Payload Filter Editor Components, continued

Attribute	Description
	 Operator Value in 4 5 matches any incident with a varbind value of either **4** or **5**. **Note**: As shown in the example, each value must be entered on a separate line. NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example (**4, 5**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line. • **is not null** Finds all non-blank values. Click here for an example. Example: `ciaValue is not null` matches any incident with a varbind that contains a value. • **is null** Finds all blank values. Click here for an example. Example: `ciaValue is null` matches any incident with a varbind that does not contain a value. • **like** Finds matches using the syntax defined for java regular expressions. See the Pattern (Java Platform SE6) API documentation at : `http://download.oracle.com/javase/6/docs/api/java/util/regex/Pattern.html` for more information. Click here for more information. The period asterisk (.*) characters mean *any number of characters of any type at this location*. The period (.) character means *any single character of any type at this location*. **Note**: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below. Example: `ciaName like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that begins with 1.3.6.1.4.1.9.9 and (optionally) ends with any number of characters. `ciaValue like .*Chicago.*` finds all traps or events that contain a

Suppression Configuration Attributes , continued

Name	Description
	Payload Filter Editor Components, continued

Attribute	Description
	varbind value that includes the string **Chicago**. - **not between** Finds all values except those between the two values specified. Click here for an example. Example: `ciaValue not between 5 8` matches an incident that contains a varbind with the values less than **5** or greater than **8** . - **not in** Finds all values except those included in the list of values. Click here for an example. Example: `ciaValue not in`  Operator: not in Value: 1 2 matches any incident that contains a varbind with values other than **1** and **2**. **Note**: As shown in the example, each value must be entered on a separate line. NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example, (**1, 2**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line. - **not like** Finds all that do not have the values specified using the syntax defined for Java regular expressions. See the Pattern (Java Platform SE6) API documentation at: `http://download.oracle.com/javase/6/docs/api/java/util/regex/Pattern.html` for more information. Click here for an example. The period asterisk (.*) characters mean *any number of characters of any type at this location*. The period (.) character means *any single character of any type at this location*. **Note**: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below.

Suppression Configuration Attributes , continued

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td></td><td>
<b>Payload Filter Editor Components, continued</b>
<table>
<tr><th>Attribute</th><th>Description</th></tr>
<tr><td></td><td>Example:
`ciaName not like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that does not begin with <b>.1.3.6.1.4.1.9.9</b> and (optionally) ends with any number of characters.
`ciaValue not like .*Chicago.*` finds all traps or events that do not contain a varbind value that includes the string <b>Chicago</b>.</td></tr>
<tr><td>Value</td><td>The value for which you want NNMi to search.
Note the following:
• The values you enter are case sensitive.
• NNMi displays a variable number of value fields depending on the Operator selected. For example, the `between` Operator causes two value fields to be displayed.
• The `between, in` and `not in` operators require that each value be entered on a separate line.</td></tr>
</table>
<b>Payload Filter Editor Buttons</b>
<table>
<tr><th>Button</th><th>Description</th></tr>
<tr><td>Append</td><td>Appends the current expression (Attribute, Operator,and Value) to the selected expression already included in the filter string.</td></tr>
<tr><td>Insert</td><td>Inserts the current expression (Attribute, Operator,and Value) in front of the cursor location within the Filter String.</td></tr>
<tr><td>Replace</td><td>Replaces the selected expression with the expression displayed in the Attribute, Operator, and Value fields.</td></tr>
<tr><td>AND</td><td>Inserts the AND Boolean Operator in the selected cursor location.
<b>Note</b>: View the expression displayed under <b>Filter String</b> to see the logic of the expression as it is created.</td></tr>
<tr><td>OR</td><td>Inserts the OR Boolean Operator in the current cursor location.
<b>Note</b>: View the expression displayed under <b>Filter String</b> to see the logic of the expression as it is created.</td></tr>
<tr><td>NOT</td><td>Can be used in any part of the Filter String to specify that NNMi should</td></tr>
</table>
</td></tr>
</table>

Suppression Configuration Attributes , continued

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td></td><td>

Payload Filter Editor Buttons, continued

<table>
<tr><th>Button</th><th>Description</th></tr>
<tr><td></td><td>exclude interfaces with values that pass the expression that immediately follows the NOT.
For example, when evaluating the following expression, NNMi includes interfaces with (interface description) ifDesc containing <b>VLAN</b>, and excludes any Interfaces that have <b>VLAN10</b> for the (interface name) ifName value:
(ifDesc like VLAN AND NOT (ifName=VLAN10))
<b>Note</b>: View the expression displayed under Filter String to see the logic of the expression as it is created .</td></tr>
<tr><td>EXISTS</td><td>Used for filters that include Capabilities or Custom Attribute names and values in the Filer String.
Indicates that you want NNMi to consider interfaces that do not have any Capabilities or Custom Attributes when evaluating the Filter String.
<b>Note</b>: If you include Capabilities or Custom Attribute names and values in the Filer String, but do not use <b>EXISTS</b> or <b>NOT EXISTS</b>, NNMi excludes from its search interfaces that do not include Capabilities or Custom Attributes.
For example, when evaluating the following Filter String, NNMi includes interfaces with (interface description) ifDesc containing <b>VLAN</b>, as well as any Interfaces Custom Attribute Role value is <b>LAN Connection to Oracle Server</b>:
(ifDesc like VLAN OR EXISTS((customAttrName=Role AND customAttrValue=LAN Connection to Oracle Server)))
<b>Note</b>: View the expression displayed under Filter String to see the logic of the expression as it is created.</td></tr>
<tr><td>NOT EXISTS</td><td>Used for filters that include Capabilities or Custom Attribute names and values in the Filer String. Indicates that you want NNMi to consider interfaces that do not have any Capabilities or Custom Attributes when evaluating the Filter String, but exclude the interfaces that match the expression that follows the <b>NOT EXISTS</b>.
<b>Note</b>: If you include Capabilities or Custom Attribute names and values in the Filer String, but do not use <b>EXISTS</b> or <b>NOT EXISTS</b>, NNMi excludes from its search interfaces that do not</td></tr>
</table>

</td></tr>
</table>

Suppression Configuration Attributes , continued

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td></td><td><b>Payload Filter Editor Buttons, continued</b>
<table>
<tr><th>Button</th><th>Description</th></tr>
<tr><td></td><td>include Capabilities or Custom Attributes.
For example, when evaluating the following expression, NNMi includes interfaces with (interface description) <code>ifDesc</code> containing <b>VLAN</b>, and excludes any Interfaces that have the Custom Attribute Role and that Role value is <b>LAN Connection to Oracle Server</b>:
<code>(ifDesc like VLAN OR NOT EXISTS ((customAttrName=Role AND customAttrValue=LAN Connection to Oracle Server)))</code>
<b>Note</b>: View the expression displayed under Filter String to see the logic of the expression as it is created.</td></tr>
<tr><td>Delete</td><td>Deletes the selected expression.
<b>Note</b>: If the Boolean Operator is selected, the Payload Filter Editor deletes all expressions associated with the Boolean Operator.</td></tr>
</table>
</td></tr>
</table>

Configure Enrichment Settings for a Remote NNM 6.x/7.x Event Incident

For information about each Remote NNM 6.x/7.x Events tab:

NNMi enables you to fine tune and enhance incidents based on Interface Group, Node Group, or default Enrichment settings. NNMi applies your Enrichment settings in the following order. Only the first match applies.

1. Interface Group (Remote 6.x/7.x Event Configuration Form: Interface Settings tab)
2. Node Group (Remote 6.x/7.x Event Configuration Form: Node Settings tab)
3. Enrichment configuration settings without specifying an Interface Group or Node Group (Remote 6.x/7.x Event Configuration Form: Enrichment tab.)

The types of items you can fine tune and enhance for a selected incident configuration, include:

- Category
- Family
- Severity
- Priority
- Correlation Nature

- Message
- Assigned To

Note: Any configuration you specify for Severity, Priority, or Message overrides those values provided in the Remote NNM 6.x/7.x Event Configuration Form: Basics information.

A Payload Filter enables you to use the data that is included with any of the following items before they are stored as incidents in NNMi:

- Traps generated from an SNMP agent
- Management incidents that are generated by NNMi
- Events generated by NNM 6.x or 7.x management stations

Examples of the type of data that can be used as a Payload Filter include SNMP trap varbind names and values as well as CIA (Custom Incident Attribute) names and values. For example, you might want NNMi to suppress a particular status change notification trap for a specified Node Group or Interface Group. To do so, you could include the name of the trap varbind that stores this information as well as the particular status change value string the traps that you want to suppress should contain.

Note: The CIA added to an incident must be provided by NNMi. You cannot create CIAs.

See "Configure Incident Enrichment Settings for an Interface Group (Remote NNM 6.x/7.x Events) " on page 1250 for information about how to enrich an incident for an Interface Group with or without a Payload Filter.

See "Configure Incident Enrichment Settings for a Node Group (Remote NNM 6.x/7.x Events)" on page 1287 for more information about how to enrich an incident for a Node Group with or without a Payload Filter.

To configure Enrich Settings for an incident using a Payload Filter without an Interface Group or Node Group Filter:

1. Navigate to the **Remote NNM 6.x/7.x Event Configuration** form:
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Expand the **Incidents** folder.
   c. Select **Remote NNM 6.x/7.x Event Configurations** .
   d. Do one of the following:
      i. To create an incident configuration, click the ✳ New icon, and continue.
      ii. To edit an incident configuration, select a row, click the 📂 Open icon, and continue.
      iii. To delete an incident configuration, select a row and click the ✖ Delete icon.
2. Select the **Enrichment** tab.
3. Do one of the following:
   a. To create a new configuration, click the ✳ New icon.
   b. To edit an existing configuration, select a row, click the 📂 Open icon, and continue.

4. Provide the required information (see table)

5. Click **Save and Close** to save your changes and return to the previous form.

Enrichment Attributes

Name	Description
Category	Use the Category attribute to customize the category for this incident configuration. Possible values include: • Accounting • Application Status • Configuration • Fault • Performance • Security • Status See "Specify Category and Family Attribute Values for Organizing Your Incidents (SNMP Trap Incident)" on page 790 for more information.
Family	Use the Family attribute to customize the Family for this incident configuration. Select from the drop-down list or create a new value. For example, some of the values provided by NNMi include: • Address • Aggregated Port (Interfaces using **Link Aggregation**[1] protocol. See Interface Form: Link Aggregation tab.) • Card • Connection • Correlation • Interface • Node
Severity	The incident Severity represents the seriousness calculated for the incident. Use the Severity attribute to specify the severity that should be assigned to the incident you are configuring. Possible values are described below:

[1] Protocols used on Switches to configure multiple Interfaces (Aggregation Member Interfaces) to function as if they were one (an Aggregator Interface). When two Aggregator Interfaces establish a connection, that connection is an Aggregator Layer 2 Connection. The Aggregator Layer 2 Connection appears on Layer 2 Neighbor View maps as a thick line with an Interface icon at each end (representing the Aggregator Interface). When you double-click the thick line, it converts into multiple thin lines representing the participating Aggregation Member Layer 2 Connections with their Aggregation Member Interfaces at each end of the lines.

Enrichment Attributes , continued

Name	Description
	Normal - Indicates there are no known problems related to the associated object. This Severity is meant to be informational. Generally, no action is needed for these incidents. **Warning** - Indicates there might be a problem related to the associated object. **Minor** - Indicates NNMi has detected problems related to the associated object that require further investigation. **Major** - Indicates NNMi has detected problems related to the associated object to be resolved before they become critical. **Critical** - Indicates NNMi has detected problems related to the associated object that require immediate attention.
Priority	Used to communicate the urgency of resolving the selected incident. You control this value. NNMi sets this value to null by default. The lower the number the higher the priority. Possible values are: 5 **None** 4 **Low** 3 **Medium** 2 **High** 1 **Top** **Note**: The icons are displayed only in table views.
Correlation Nature	Use the Correlation Nature to customize the Correlation Nature for this incident configuration. Possible values include: • Info • None • Root Cause • Secondary Root Cause • Symptom • Stream Correlation • Service Impact • Dedup Stream Correlation • Rate Stream Correlation See Incident Form: General Tab for more information.

Enrichment Attributes , continued

Name	Description
Message Format	When configuring an incident, specify how the incident message appears in the incident view. The string you specify in the Message Format attribute is visible in an incident view. **Note**: The incident message limit is 1024 characters. If you exceed this limit, NNMi truncates the value starting from the right. You can use any combination of default and custom attributes: "Valid Parameters for Configuring Incident Messages (SNMP Trap Incident)" on page 795 "Include Custom Incident Attributes in Your Message Format (SNMP Trap Incident)" on page 801
Assigned To	Use to specify the owner of any incident generated for this incident configuration. Click the Lookup icon and select Quick Find to select a valid user name. **Note**: You can also begin to type a valid user name and use the auto-complete feature to select the user name of interest.
Description	Use the Description attribute to provide additional information that you want to note about the current enhancement configuration. This description applies only to the enhancement configuration and does not appear when NNMi displays any associated incident. Type a maximum of 1024 characters. Alpha-numeric and special characters (~ ! @ # $ % ^ & * ( ) _+ -) are permitted.

Configure Dampening Settings for a Remote NNM 6.x/7.x Event Incident

For information about each Remote NNM 6.x/7.x Events tab:

NNMi enables you to delay the following for an incident configuration based on the Source Object's participation in an Interface Group:

- Execution of Incident Actions
- Execution of Diagnostics (*HP Network Node Manager iSPI Network Engineering Toolset Software \ NNM iSPI NET*)
- Appearance within Incident views in the NNMi Console

You can dampen incidents based on Interface Group, Node Group, or default Dampening settings. NNMi applies your Dampening settings in the following order. Only the first match applies.

1. Interface Group (Remote 6.x/7.x Event Configuration Form: Interface Settings tab)
2. Node Group (Remote 6.x/7.x Event Configuration Form: Node Settings tab)

3. Dampening settings without specifying an Interface Group or Node Group (Remote 6.x/7.x Event Configuration Form: Dampening tab)

When using the Dampening configuration, note the following:

- Duplicate and Rate Correlation incidents inherit the Dampening settings from its Correlated Children. If the Correlated Children are Closed while Dampened, and therefore deleted, NNMi retains the parent Duplicate or Rate Correlation incident. See "Correlate Duplicate Incidents (Deduplication Configuration)" on page 659 and "Track Incident Frequency (Rate: Time Period and Count)" on page 659 for more information about Duplicate and Rate Correlation incidents.

  Note: NNMi administrators can view the number of incidents Closed and deleted while dampened. Access the **Help** → **System Information** → **Health** tab, click the View Detailed Health Report button, and search for the word dampened.

- After the Dampen Interval has passed, NNMi changes the Lifecycle State to REGISTERED.

  See About the Incident Lifecycle for more information about Lifecycle State.

See "Configure Incident Dampening Settings for an Interface Group (Remote NNM 6.x/7.x Events)" on page 1262 for information about how to configure Dampening for an Interface Group with or without a Payload Filter.

See "Configure Incident Dampening Settings for a Node Group (Remote NNM 6.x/7.x Events)" on page 1299 for more information about how to configure Dampening for a Node Group with or without a Payload Filter.

To configure Dampening settings for an incident using a Payload Filter without an Interface Group or Node Group Filter:

1. Navigate to the **Remote NNM 6.x/7.x Event Configuration** form:
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Expand the **Incidents** folder.
   c. Select **Remote NNM 6.x/7.x Event Configurations** .
   d. Do one of the following:
      i. To create a configuration, click the New icon, and continue.
      ii. To edit configuration, select a row, click the Open icon, and continue.
      iii. To delete a configuration, select a row and click the Delete icon.
2. Select the **Dampening** tab.
3. Provide the required information (see table)
4. Click **Save and Close** to save your changes and return to the previous form.

Dampening Attributes

Name	Description
Enable	Use this attribute to temporarily disable an incident's Dampening settings.

Dampening Attributes , continued

Name	Description
	To temporarily disable the Dampening settings for the selected incident configuration, clear **Enable** □. To enable the Dampening settings for the selected incident configuration, click **Enable** ☑.
Hour	Specifies the number of hours to be used for the Dampen Interval.
Minutes	Specifies the number of minutes to be used for the Dampen Interval.
Seconds	Specifies the number of seconds to be used for the Dampen Interval.

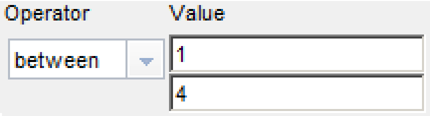
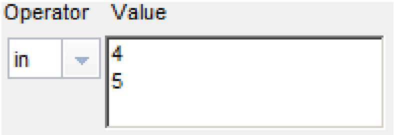
Payload Filter	The Payload Filter Editor enables you to create expressions that further refine the filters used to select the incidents to be suppressed, enriched, or dampened. Make sure to design any complex Payload Filters offline as a Boolean expression first. This method can help to minimize errors when entering your expressions using the Payload Filter editor.
	When creating a Payload Filter, note the following:
	▪ Payload Filter expressions for the `like` and `not like` operators use the syntax defined for java regular expressions (java.util.regex Pattern class).
	▪ You must use a `ciaName` that already exists in the trap or event you are configuring.
	▪ Each set of expressions associated with a Boolean Operator is treated as if it were enclosed in parentheses and evaluated together.
	▪ View the expression displayed under **Filter String** to see the logic of the expression as it is created.
	▪ The `AND` and `OR` Boolean Operators must contain at least two expressions as shown in the example below.
	The following example filters incidents on voltage state:
	`AND`
	`    ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7`
	`    ciaValue = 5`
	NNMi evaluates the expression above as follows:
	`(ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7 AND ciaValue = 5)`
	NNMi finds all incidents with a varbind `.1.3.6.1.4.1.9.9.13.1.2.1.7` value of **5**.
	▪ The placement of your cursor and the subsequent text that is selected is important when performing operations using the Payload Filter Editor. For example, you

Dampening Attributes , continued

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td></td><td>
append to, replace, or change the indentation of the expression that is selected.
<ul>
<li>The placement of your cursor and the subsequent text that is selected is especially important when adding your Boolean operators.</li>
<li>You can include more than one varbind in the same Payload Filter expression as shown in the following example:
<pre>((ciaName like \Q.1.3.6.1.4.1.9.9\E.* AND ciaValue = 25)
AND (ciaName like \Q.1.3.6.1.2.1.2.2.1.1.3\E.* AND
ciaValue = 3))</pre>
In this example, a given trap must meet each of the following criteria:
<ul>
<li>Contain a varbind whose Object Identifier (OID) matches the regular expression <code>\Q.1.3.6.1.4.1.9.9\E.*</code> and has a value of <code>25</code>.</li>
<li>Contain a varbind whose OID matches the regular expression <code>\Q.1.3.6.1.2.1.2.2.1.1.3\E.*</code> and has a value of <code>3</code>.</li>
</ul>
</li>
</ul>
<b>Payload Filter Editor Components</b>
<table>
<tr><th>Attribute</th><th>Description</th></tr>
<tr><td>Attribute</td><td>The attribute name on which NNMi searches. Filterable attributes include the following:
<ul>
<li>ciaName</li>
<li>ciaValue</li>
</ul></td></tr>
<tr><td>Operator</td><td>Valid operators are described below.
<ul>
<li><b>=</b> Finds all values equal to the value specified. Click here for an example.
Example: <code>ciaName=.1.3.6.1.4.1.9.9.13.1.2.1.7</code> matches any incident that contains a varbind with the name value of <b>.1.3.6.1.4.1.9.9.13.1.2.1.7</b>.</li>
<li><b>!=</b> Finds all values not equal to the value specified. Click here for an example.
Example: <code>ciaName! = .1.3.6.1.4.1.9.9.13.1.2.1.7</code> matches any incident that contains a varbind with a name value other than <b>1.3.6.1.4.1.9.9.13.1.2.1.7</b>.</li>
<li><b>&lt;</b> Finds all values less than the value specified. Click here for an example.
Example: <code>ciaValue &lt; 6</code> matches any incident that contains a</li>
</ul></td></tr>
</table>
</td></tr>
</table>

Dampening Attributes , continued

Name	Description
	Payload Filter Editor Components, continued

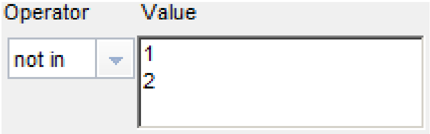
Attribute	Description
	varbind with a value less than **6**. - **<=** Finds all values less than or equal to the value specified. Click here for an example. Example: `ciaValue <= 6` matches any incident that contains a varbind with a value less than or equal to **6**. - **>** Finds all values greater than the value specified. Click here for an example. Example: `ciaValue > 4` matches any incident that contains a varbind with a value greater than **4**. - **>=** Finds all values greater than or equal to the value specified. Click here for an example. Example: `ciaValue >= 4` matches any incident that contains a varbind with values greater than or equal to **4**. - **between** Finds all values equal to and between the two values specified. Click here for an example. Example: `ciaValue between`  Operator: between Value: 1 4 matches any incident that contains a varbind value equal to or greater than **1** and equal to or less than **4**. **Note**: As shown in the example, each value must be entered on a separate line. - **in** Finds any match to at least one value in a list of values. Click here for an example. Example: `ciaValue in`  Operator: in Value: 4 5 matches any incident with a varbind value of either **4** or **5**.

Dampening Attributes , continued

Name	Description
	Payload Filter Editor Components, continued

Attribute	Description
	Note: As shown in the example, each value must be entered on a separate line. NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example (**4, 5**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line. ▪ **is not null** Finds all non-blank values. Click here for an example. Example: `ciaValue is not null` matches any incident with a varbind that contains a value. ▪ **is null** Finds all blank values. Click here for an example. Example: `ciaValue is null` matches any incident with a varbind that does not contain a value. ▪ **like** Finds matches using the syntax defined for java regular expressions. See the Pattern (Java Platform SE6) API documentation at : `http://download.oracle.com/javase/6/docs/api/java/util/regex/Pattern.html` for more information. Click here for more information. The period asterisk (.*) characters mean *any number of characters of any type at this location*. The period (.) character means *any single character of any type at this location*. **Note**: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below. Example: `ciaName like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that begins with 1.3.6.1.4.1.9.9 and (optionally) ends with any number of characters. `ciaValue like .*Chicago.*` finds all traps or events that contain a varbind value that includes the string **Chicago**. ▪ **not between** Finds all values except those between the two values specified. Click here for an example.

Dampening Attributes , continued

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td></td><td>
<b>Payload Filter Editor Components, continued</b>
<table>
<tr><th>Attribute</th><th>Description</th></tr>
<tr><td></td><td>
Example: <code>ciaValue not between 5 8</code> matches an incident that contains a varbind with the values less than <b>5</b> or greater than <b>8</b> .
▪ <b>not in</b> Finds all values except those included in the list of values. Click here for an example.
Example:
<code>ciaValue not in</code>

Operator: not in
Value: 1
2
matches any incident that contains a varbind with values other than <b>1</b> and <b>2</b>.
<b>Note</b>: As shown in the example, each value must be entered on a separate line.
NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example, (<b>1, 2</b>). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line.
▪ <b>not like</b> Finds all that do not have the values specified using the syntax defined for Java regular expressions. See the Pattern (Java Platform SE6) API documentation at: <code>http://download.oracle.com/javase/6/docs/api/java/util/regex/Pattern.html</code> for more information. Click here for an example.
The period asterisk (.*) characters mean <i>any number of characters of any type at this location</i>.
The period (.) character means <i>any single character of any type at this location</i>.
<b>Note</b>: To include literal string values in the Value attribute, enclose the string value in \Q&lt;<i>literal_value</i>&gt;\E as shown in the Examples listed below.
Example:
<code>ciaName not like \Q.1.3.6.1.4.1.9.9\E.*</code> matches any
</td></tr>
</table>
</td></tr>
</table>

Dampening Attributes , continued

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td></td><td>

Payload Filter Editor Components, continued

<table>
<tr><th>Attribute</th><th>Description</th></tr>
<tr><td></td><td>incident that contains a varbind name value that does not begin with <b>.1.3.6.1.4.1.9.9</b> and (optionally) ends with any number of characters.
<code>ciaValue not like .*Chicago.*</code> finds all traps or events that do not contain a varbind value that includes the string <b>Chicago</b>.</td></tr>
<tr><td>Value</td><td>The value for which you want NNMi to search.
Note the following:
▪ The values you enter are case sensitive.
▪ NNMi displays a variable number of value fields depending on the Operator selected. For example, the <code>between</code> Operator causes two value fields to be displayed.
▪ The <code>between, in</code> and <code>not in</code> operators require that each value be entered on a separate line.</td></tr>
</table>

Payload Filter Editor Buttons

<table>
<tr><th>Button</th><th>Description</th></tr>
<tr><td>Append</td><td>Appends the current expression (Attribute, Operator,and Value) to the selected expression already included in the filter string.</td></tr>
<tr><td>Insert</td><td>Inserts the current expression (Attribute, Operator,and Value) in front of the cursor location within the Filter String.</td></tr>
<tr><td>Replace</td><td>Replaces the selected expression with the expression displayed in the Attribute, Operator, and Value fields.</td></tr>
<tr><td>AND</td><td>Inserts the AND Boolean Operator in the selected cursor location.
<b>Note</b>: View the expression displayed under <b>Filter String</b> to see the logic of the expression as it is created.</td></tr>
<tr><td>OR</td><td>Inserts the OR Boolean Operator in the current cursor location.
<b>Note</b>: View the expression displayed under <b>Filter String</b> to see the logic of the expression as it is created.</td></tr>
<tr><td>NOT</td><td>Can be used in any part of the Filter String to specify that NNMi should exclude interfaces with values that pass the expression that immediately follows the <code>NOT</code>.</td></tr>
</table>

</td></tr>
</table>

Dampening Attributes , continued

Nam e	Description
	Payload Filter Editor Buttons, continued

Button	Description
	For example, when evaluating the following expression, NNMi includes interfaces with (interface description) `ifDesc` containing **VLAN**, and excludes any Interfaces that have **VLAN10** for the (interface name) `ifName` value: `(ifDesc like VLAN AND NOT (ifName=VLAN10))` **Note**: View the expression displayed under Filter String to see the logic of the expression as it is created .
EXISTS	Used for filters that include Capabilities or Custom Attribute names and values in the Filer String. Indicates that you want NNMi to consider interfaces that do not have any Capabilities or Custom Attributes when evaluating the Filter String. **Note**: If you include Capabilities or Custom Attribute names and values in the Filer String, but do not use **EXISTS** or **NOT EXISTS**, NNMi excludes from its search interfaces that do not include Capabilities or Custom Attributes. For example, when evaluating the following Filter String, NNMi includes interfaces with (interface description) `ifDesc` containing **VLAN**, as well as any Interfaces Custom Attribute Role value is **LAN Connection to Oracle Server**: `(ifDesc like VLAN OR EXISTS ((customAttrName=Role AND customAttrValue=LAN Connection to Oracle Server)))` **Note**: View the expression displayed under Filter String to see the logic of the expression as it is created.
NOT EXISTS	Used for filters that include Capabilities or Custom Attribute names and values in the Filer String. Indicates that you want NNMi to consider interfaces that do not have any Capabilities or Custom Attributes when evaluating the Filter String, but exclude the interfaces that match the expression that follows the **NOT EXISTS**. **Note**: If you include Capabilities or Custom Attribute names and values in the Filer String, but do not use **EXISTS** or **NOT EXISTS**, NNMi excludes from its search interfaces that do not include Capabilities or Custom Attributes.

Dampening Attributes , continued

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td></td><td>
<b>Payload Filter Editor Buttons, continued</b>
<table>
<tr><th>Button</th><th>Description</th></tr>
<tr><td></td><td>For example, when evaluating the following expression, NNMi includes interfaces with (interface description) ifDesc containing <b>VLAN</b>, and excludes any Interfaces that have the Custom Attribute Role and that Role value is <b>LAN Connection to Oracle Server</b>:
(ifDesc like VLAN OR NOT EXISTS ((customAttrName=Role AND customAttrValue=LAN Connection to Oracle Server)))
<b>Note</b>: View the expression displayed under Filter String to see the logic of the expression as it is created.</td></tr>
<tr><td>Delete</td><td>Deletes the selected expression.
<b>Note</b>: If the Boolean Operator is selected, the Payload Filter Editor deletes all expressions associated with the Boolean Operator.</td></tr>
</table>
</td></tr>
</table>

Configure Forward to Global Managers Settings for a Remote 6.x/7.x Event Incident (*NNMi Advanced*)

For information about each Remote 6.x/7.x Events tab:

(*NNMi Advanced - Global Network Management feature*) The NNMi Global Network Management feature enables multiple NNMi management servers to work together while managing different geographic areas of your network. See NNMi's Feature (NNMi Advanced) for more information. The Global Manager combines topology information from multiple Regional Managers, but maintains *a separate set of incidents about those nodes*.

Use the Global Manager Forwarding tab when you want to forward specific NNM 6.x/7.x Events from your NNMi management server (a Regional Manager) to all Global Managers in your Global Network Management environment.

Caution: The Global Manager must have an incident configuration for that NNM 6.x/7.x Event, otherwise the in-coming NNM 6.x/7.x Event is dropped. See "Export and Import Configuration Settings" on page 1579 for ideas about sharing incident configurations among NNMi management servers.

When you configure Forward to Global Managers, you can specify an optional Payload Filter for NNMi to use when determining *which occurrences* should be forwarded to Global Managers. Payload Filters enables you to use the data that is included with an occurrence of an incident configuration before it is stored as an incident in the NNMi database.

Examples of the type of data that can be used as a Payload Filter include Custom Incident Attribute names (ciaName) and values (ciaValue). For example, you might want NNMi to forward an incident based on a particular status change notification trap. To do so, you would specify a Payload Filter that includes the name of the Custom Incident Attribute that stores the status information as well as the status change value string of interest.

To configure forwarding to Global Managers:

1. Navigate to the **Remote NNM 6.x/7.x Event Configuration** form:
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Expand the **Incidents** folder.
   c. Select **Remote NNM 6.x/7.x Event Configurations** .
   d. Do one of the following:
      i. To create an incident configuration, click the New icon, and continue.
      ii. To edit an incident configuration, select a row, click the Open icon, and continue.
      iii. To delete an incident configuration, select a row and click the Delete icon.
2. Select the **Forward to Global Managers** tab.
3. Provide the required information (see table)
4. Click **Save and Close** to save your changes and return to the previous form.

Forwarding Configuration Attributes

Name	Description
Enable	Use this attribute to enable or temporarily disable an incident's GNM settings. To temporarily disable the GNM Forwarding Configuration settings for the selected incident configuration, clear **Enable** □. To enable the GNM Forwarding settings for the selected incident configuration, click **Enable** ☑.
Payload Filter	The Payload Filter Editor enables you to create expressions that further refine the filters used to select the incidents that NNMi forwards to other servers. Make sure to design any complex Payload Filters offline as a Boolean expression first. This method can help to minimize errors when entering your expressions using the Payload Filter editor. When creating a Payload Filter, note the following: • Payload Filter expressions for the `like` and `not like` operators use the syntax defined for java regular expressions (java.util.regex Pattern class). • You must use a `ciaName` that already exists in the trap or event you are configuring. • Each set of expressions associated with a Boolean Operator is treated as if it were enclosed in parentheses and evaluated together. • View the expression displayed under **Filter String** to see the logic of the

Forwarding Configuration Attributes , continued

Name	Description

	expression as it is created.
	• The `AND` and `OR` Boolean Operators must contain at least two expressions as shown in the example below.
	The following example filters incidents on voltage state:
	`AND`
	`      ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7`
	`      ciaValue = 5`
	NNMi evaluates the expression above as follows:
	`(ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7 AND ciaValue = 5)`
	NNMi finds all incidents with a varbind value of `.1.3.6.1.4.1.9.9.13.1.2.1.7` and CIA value of **5**.
	• The placement of your cursor and the subsequent text that is selected is important when performing operations using the Payload Filter Editor. For example, you append to, replace, or change the indentation of the expression that is selected.
	• The placement of your cursor and the subsequent text that is selected is especially important when adding your Boolean operators.

Payload Filter Editor Components

Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following: • ciaName • ciaValue
Operator	Valid operators are described below. • **=** Finds all values equal to the value specified. Click here for an example. Example: `ciaName=.1.3.6.1.4.1.9.9.13.1.2.1.7` matches any incident that contains a varbind with the name value **.1.3.6.1.4.1.9.9.13.1.2.1.7**. • **!=** Finds all values not equal to the value specified. Click here for an example. Example: `ciaName! = .1.3.6.1.4.1.9.9.13.1.2.1.7` matches any incident that contains a varbind with the name value other than **1.3.6.1.4.1.9.9.13.1.2.1.7**. • **<** Finds all values less than the value specified. Click here for an example.

Forwarding Configuration Attributes , continued

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td></td><td>

Payload Filter Editor Components, continued

<table>
<tr><th>Attribute</th><th>Description</th></tr>
<tr><td></td><td>

Example: `ciaValue < 6` matches any incident that contains a varbind value less than **6**.

- **<=** Finds all values less than or equal to the value specified. Click here for an example.

  Example: `ciaValue <= 6` matches any incident that contains a varbind value less than or equal to **6**.

- **>** Finds all values greater than the value specified. Click here for an example.

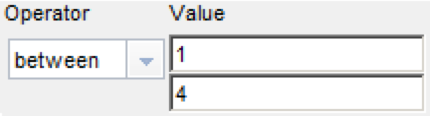
  Example: `ciaValue > 4` matches any incident that contains a varbind value greater than **4**.

- **>=** Finds all values greater than or equal to the value specified. Click here for an example.

  Example: `ciaValue >= 4` matches any incident that contains a varbind with values greater than or equal to **4**.

- **between** Finds all values equal to and between the two values specified. Click here for an example.

  Example: `ciaValue between`

  Operator: between
  Value: 1
  4

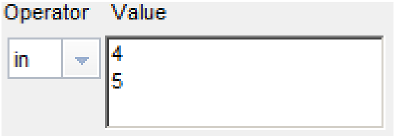
  matches any incident that contains a varbind value equal to or greater than **1** and equal to or less than **4**.

  Note: As shown in the example, each value must be entered on a separate line.

- **in** Finds any match to at least one value in a list of values. Click here for an example.

  Example:

  `ciaValue in`

  Operator: in
  Value: 4
  5

  matches any incident that contains a varbind value of either **4** or **5**.

</td></tr>
</table>

</td></tr>
</table>

Forwarding Configuration Attributes , continued

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td></td><td>
<b>Payload Filter Editor Components, continued</b>
<table>
<tr><th>Attribute</th><th>Description</th></tr>
<tr><td></td><td>
<b>Note</b>: As shown in the example, each value must be entered on a separate line.
NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example (<b>4, 5</b>). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line.
• <b>is not null</b> Finds all non-blank values. Click here for an example.
Example: <code>ciaValue is not null</code> matches any incident with a varbind value.
• <b>is null</b> Finds all blank values. Click here for an example.
Example: <code>ciaValue is null</code> matches any incident with no varbind values.
• <b>like</b> Finds matches using wildcard characters. Click here for more information about using wildcard characters.
The period asterisk (.*) characters mean <i>any number of characters of any type at this location</i>.
The period (.) character means <i>any single character of any type at this location</i>.
<b>Note</b>: To include literal string values in the Value attribute, enclose the string value in \Q&lt;<i>literal_value</i>&gt;\E as shown in the Examples listed below.
Example:
<code>ciaName like \Q.1.3.6.1.4.1.9.9\E.*</code> matches any incident that contains a varbind name value that begins with 1.3.6.1.4.1.9.9 and (optionally) ends with any number of characters.
<code>ciaValue like .*Chicago.*</code> finds all traps or events that contain a varbind value that includes the string <b>Chicago</b>.
• <b>not between</b> Finds all values except those between the two values specified. Click here for an example.
Example: <code>ciaValue not between 5 8</code> matches an incident that contains a varbind with the values less than <b>5</b> or greater than <b>8</b> .
• <b>not in</b> Finds all values except those included in the list of values. Click here for an example.
Example:
</td></tr>
</table>
</td></tr>
</table>

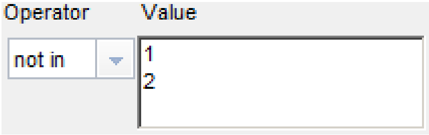
Forwarding Configuration Attributes , continued

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td></td><td>

Payload Filter Editor Components, continued

<table>
<tr><th>Attribute</th><th>Description</th></tr>
<tr><td></td><td>

`ciaValue not in`



Operator: not in
Value:
1
2

matches any incident that contains a varbind with values other than **1** and **2**.

Note: As shown in the example, each value must be entered on a separate line.

NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example, (**1, 2**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line.

- **not like** Finds all that do not have the values specified (using wildcard strings). Click here for an example.

The period asterisk (.*) characters mean *any number of characters of any type at this location*.

The period (.) character means *any single character of any type at this location*.

Note: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below.

Example:

`ciaName not like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that does not begin with **.1.3.6.1.4.1.9.9** and (optionally) ends with any number of characters.

`ciaValue not like .*Chicago.*` finds all traps or events that do not contain a varbind value that includes the string **Chicago**.

</td></tr>
<tr><td>Value</td><td>

The value for which you want NNMi to search.

Note the following:

- The values you enter are case sensitive.
- NNMi displays a variable number of value fields depending on the Operator selected. For example, the `between` Operator causes two value fields to be displayed.

</td></tr>
</table>

</td></tr>
</table>

Forwarding Configuration Attributes , continued

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td></td><td>

Payload Filter Editor Components, continued

<table>
<tr><th>Attribute</th><th>Description</th></tr>
<tr><td></td><td>

- The `between, in` and `not in` operators require that each value be entered on a separate line.

</td></tr>
</table>

Payload Filter Editor Buttons

<table>
<tr><th>Button</th><th>Description</th></tr>
<tr><td>Append</td><td>Appends the current expression (Attribute, Operator,and Value) to the selected expression already included in the filter string.</td></tr>
<tr><td>Insert</td><td>Inserts the current expression (Attribute, Operator,and Value) in front of the cursor location within the Filter String.</td></tr>
<tr><td>Replace</td><td>Replaces the selected expression with the expression displayed in the Attribute, Operator, and Value fields.</td></tr>
<tr><td>AND</td><td>

Inserts the AND Boolean Operator in the selected cursor location.

Note: View the expression displayed under **Filter String** to see the logic of the expression as it is created.

</td></tr>
<tr><td>OR</td><td>

Inserts the OR Boolean Operator in the current cursor location.

Note: View the expression displayed under **Filter String** to see the logic of the expression as it is created.

</td></tr>
<tr><td>NOT</td><td>

Can be used in any part of the Filter String to specify that NNMi should exclude interfaces with values that pass the expression that immediately follows the `NOT`.

For example, when evaluating the following expression, NNMi includes interfaces with (interface description) `ifDesc` containing **VLAN**, and excludes any Interfaces that have **VLAN10** for the (interface name) `ifName` value:

`(ifDesc like VLAN AND NOT (ifName=VLAN10))`

Note: View the expression displayed under Filter String to see the logic of the expression as it is created.

</td></tr>
<tr><td>EXISTS</td><td>

Used for filters that include Capabilities or Custom Attribute names and values in the Filer String.

Indicates that you want NNMi to consider interfaces that do not have any Capabilities or Custom Attributes when evaluating the Filter String.

Note: If you include Capabilities or Custom Attribute names and

</td></tr>
</table>

</td></tr>
</table>

Forwarding Configuration Attributes , continued

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td></td><td>

Payload Filter Editor Buttons, continued

<table>
<tr><th>Button</th><th>Description</th></tr>
<tr><td></td><td>

values in the Filer String, but do not use **EXISTS** or **NOT EXISTS**, NNMi excludes from its search interfaces that do not include Capabilities or Custom Attributes.

For example, when evaluating the following Filter String, NNMi includes interfaces with (interface description) `ifDesc` containing **VLAN**, as well as any Interfaces Custom Attribute Role value is **LAN Connection to Oracle Server**:

```
(ifDesc like VLAN OR EXISTS((customAttrName=Role
AND customAttrValue=LAN Connection to Oracle
Server)))
```

Note: View the expression displayed under Filter String to see the logic of the expression as it is created.

</td></tr>
<tr><td>NOT EXISTS</td><td>

Used for filters that include Capabilities or Custom Attribute names and values in the Filer String. Indicates that you want NNMi to consider interfaces that do not have any Capabilities or Custom Attributes when evaluating the Filter String, but exclude the interfaces that match the expression that follows the **NOT EXISTS**.

Note: If you include Capabilities or Custom Attribute names and values in the Filer String, but do not use **EXISTS** or **NOT EXISTS**, NNMi excludes from its search interfaces that do not include Capabilities or Custom Attributes.

For example, when evaluating the following expression, NNMi includes interfaces with (interface description) `ifDesc` containing **VLAN**, and excludes any Interfaces that have the Custom Attribute Role and that Role value is **LAN Connection to Oracle Server**:

```
(ifDesc like VLAN OR NOT EXISTS
((customAttrName=Role AND customAttrValue=LAN
Connection to Oracle Server)))
```

Note: View the expression displayed under Filter String to see the logic of the expression as it is created.

</td></tr>
<tr><td>Delete</td><td>

Deletes the selected expression.

Note: If the Boolean Operator is selected, the Payload Filter Editor deletes all expressions associated with the Boolean Operator.

</td></tr>
</table>

</td></tr>
</table>

Configure Deduplication for a Remote NNM 6.x/7.x Event Incident

For information about each Remote 6.x/7.x Events tab:

The deduplication configuration determines what values NNMi should match to detect when an SNMP trap, management event, or remote NNM 6.x/7.x event is a duplicate.

Note the following:

- Suppression, Enrichment, and Dampening are not supported for Deduplication incidents.
- NNMi applies only one deduplication configuration per incident . If NNMi generates an incident using a specified deduplication configuration, NNMi continues to correlate duplicate incidents using the original configuration. To use a different deduplication configuration for an incident, first delete the current deduplication incident (created using the original deduplication configuration). NNMi generates the next deduplication incident according to the new deduplication configuration settings.
- By default, NNMi updates the **Duplicate Count** every 30 seconds. This interval cannot be changed.
- NNMi continues to update the duplicate count regardless of an incident's lifecycle state. For example, if an incident's **Lifecycle State** is set to **Closed**, the duplicate count continues to be incremented. See About the Incident Lifecycle for more information. This behavior helps you identify situations in which the incident is not yet fixed. Take note if the Duplicate Count is incremented after a lengthy time period has elapsed, which might indicate there is a new problem with the node, interface, or address.
- Each time you stop and restart ovjboss, any incidents that have not yet been correlated or persisted are lost. This means that after a restart of ovjboss, an incoming incident might not be correlated as expected. For example, after a restart of ovjboss, a duplicate incident might not be correlated under its original parent incident. Instead, a new parent incident might be generated. See "Stop or Start an NNMi Process" on page 82for more information about starting and stopping the ovjboss process.
- If a Duplicate Correlation Incident is dampened, note the following:
  - Duplicate Correlation Incidents inherit the Dampening settings from its Correlated Children.
  - NNMi always retains the Parent Duplicate Correlation incident, even if its Child Incidents are Closed and subsequently deleted.

    See "Dampening Incident Configurations" on page 679 for more information about Dampening an incident configuration.

To specify or delete a deduplication configuration:

1. Navigate to the **Remote NNM 6.x/7.x Event Configuration** form:
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Click to expand the **Incidents** folder.
   c. Select **Remote NNM 6.x/7.x Event Cons** tab.

d. Do one of the following:

   i. To create a deduplication configuration, click the New icon, and continue.

   ii. To edit a deduplication configuration, select a row, click the Open icon, and continue.

   iii. To delete a deduplication configuration, select a row and click the Delete icon.

2. Select the **Deduplication** tab.
3. Provide the required information (see "Deduplication Attributes" table).
4. Click **Save and Close** to save your changes and return to the previous form.

Deduplication Attributes

Name	Description
Enabled	Use this attribute to temporarily disable an incident's deduplication configuration: **Enable** □ = Temporarily disable the selected configuration. **Enable** ☑ = Enable the selected configuration. **Note:** After a deduplication configuration is enabled, NNMi increments the **Duplicate Count** for an associated incident regardless of the **Lifecycle State** value. For example, if an incident's **Lifecycle State** is set to **Closed**, the duplicate count continues to be incremented. See About the Incident Lifecycle for more information.
Count	Specifies the number of duplicate incidents for the current configuration that NNMi stores at one time. For example, if the Count is 10, after NNMi receives 10 duplicate incidents, NNMi deletes the first (oldest) duplicate incident and keeps the eleventh. (NNMi stores ten maximum.)
Hours	Used with the Minute and Second Intervals to specify the time that must elapse before a new duplicate incident is generated for this incident configuration. For example, if the Hour Interval value is 1, and no Minute or Second Intervals are specified, and the duplicate incident is not generated within one hour, NNMi generates a new duplicate incident the next time it occurs.
Minutes	Used with the Hour and Second interval to specify the time that must elapse before a new duplicate incident is generated for this incident configuration. For example, if the Minute Interval is 30 and no Hour or Second Intervals are specified, and the duplicate incident is not generated within 30 minutes, NNMi generates a new duplicate incident the next time it occurs.
Seconds	Used with the Hour and Minute Intervals to specify the time that must elapse before a new duplicate incident is generated for this incident configuration. For example, if the Second Interval is 120 and no Hour or Minute Intervals are specified, and the duplicate incident is not generated within 120 seconds, NNMi generates a new duplicate incident the next time it occurs.

Deduplication Attributes, continued

Name	Description
Parent Incident	varUsed to specify the Incident Configuration that will be the Parent Incident for the incident you are configuring. For example, you might have created a Management Event Incident Configuration that could be used as the **Parent Incident** for SNMP Trap Incidents. When specifying the **Parent Incident**, you have the following options: • When you want to use a configuration that NNMi provides, use the default **Duplicate Correlation** incident configuration . If you select this option, the incident message for the Parent Incident begins as follows: `Duplicate Correlation for` *incident_configuration_name>* For example if you are configuring a **Node Down** incident and select **Duplicate Correlation** as the **Parent Incident**, the Parent Incident message begins with: **Duplicate Correlation for Node Down**. Each **Node Down** incident that is a duplicate then appears correlated under the **Duplicate Correlation for Node Down** incident. • NNMi also enables you to customize the Parent Incident for a given deduplication scenario. If you have created a Management Event Incident Configuration to use for this deduplication scenario, select the Management Event Incident Configuration that you have created.
Comparison Criteria	Specify the attribute values that must match before the incident is identified as a duplicate. The possible attributes consist of the following choices. • **Name** - The **Name** attribute value from the Incident form: General tab. • **CIA** - Represents any of the following items configured as a Parameter Value using the "Deduplication Comparison Parameters Form " on page 659: ▪ The **Value** attribute from the Incident form: Custom Attributes tab ▪ An SNMP varbind Object ID ▪ An SNMP varbind position number If you want to use CIA as part of your comparison criteria, see "Deduplication Comparison Parameters Form " on page 659 • **SourceNode** - The **Source Node** attribute value from the Basics attributes listed on the Incident form. The Source Node value is the IP Address or Name of the node for which the incident was generated. **Note**: The Source Node must be stored in the NNMi database. • **Source Object** - The **Source Object** attribute value from the Basics attributes listed on the Incident form. **Note**: The Source Object must be stored in the NNMi database. **Note**: Each attribute value in the option you select must match before the incident is identified as a duplicate. For example, if you select **Name**, only

Deduplication Attributes, continued

Name	Description
	the Incident Name value must match. If you select **Name SourceNode SourceObject CIA**, the Incident Name, Source Node, Source Object, and all Custom Incident Attribute values that you configure as a Parameter Value must match before NNMi identifies the incident as a duplicate. Selecting an option that includes CIA enables you to further refine the deduplication criteria. For example, you might want to configure deduplication for incidents with CIA values that specify the same State attribute value for a particular network object. For a description of each Comparison Criteria option, click here.

Comparison Criteria	Description
Name	Value of the **Name** attribute from the Incident form: General tab must match.
Name CIA	Each of the following values must match: • **Name** attribute from the Incident form: General tab • **CIA** - Represents the Value associated with any of the following items configured as a Parameter Value using the "Deduplication Comparison Parameters Form " on page 659: ▪ Name of a Custom Incident Attribute (CIA) provided by NNMi. (See the Incident form: Custom Attributes tab ▪ An SNMP varbind Object ID ▪ An SNMP varbind position number If you want to use CIA as part of your comparison criteria, see "Deduplication Comparison Parameters Form " on page 659
Name SourceNode	**Note**: Select this option only if the Source Node is stored in the NNMi database. Each of the following values must match: • **Name** attribute from the Incident form: General tab • The **Source Node** attribute value from the Basics attributes listed on the Incident form
Name SourceNode CIA	**Note**: Select this option only if the Source Node is stored in the NNMi database. Each of the following values must match:

Deduplication Attributes, continued

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td></td><td>
<table>
<tr><th>Comparison Criteria</th><th>Description</th></tr>
<tr><td></td><td>• **Name** attribute from the Incident form: General tab
• The **Source Node** attribute value from the Basics attributes listed on the Incident form
• **CIA** - Represents the Value associated with any of the following items configured as a Parameter Value using the "Deduplication Comparison Parameters Form " on page 659:
▪ The **Value** attribute from the Incident form: Custom Attributes tab
▪ An SNMP varbind Object ID
▪ An SNMP varbind position number
If you want to use CIA as part of your comparison criteria, see "Deduplication Comparison Parameters Form " on page 659</td></tr>
<tr><td>Name SourceObject</td><td>**Note**: Select this option only if the Source Object is stored in the NNMi database.
Each of the following values must match:
• **Name** attribute from the Incident form: General tab
• The **Source Object** attribute value from the Basics attributes listed on the Incident form.</td></tr>
<tr><td>Name SourceObject CIA</td><td>**Note**: Select this option only if the Source Object is stored in the NNMi database.
Each of the following values must match:
• **Name** attribute from the Incident form: General tab
• The **Source Object** attribute value from the Basics attributes listed on the Incident form
• **CIA** - Represents the Value associated with any of the following items configured as a Parameter Value using the "Deduplication Comparison Parameters Form " on page 659:
▪ The **Name** attribute from the Incident form: Custom Attributes tab
▪ An SNMP varbind Object ID</td></tr>
</table>
</td></tr>
</table>

Deduplication Attributes, continued

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td></td><td>
<table>
<tr><th>Comparison Criteria</th><th>Description</th></tr>
<tr><td></td><td>- An SNMP varbind position number
If you want to use CIA as part of your comparison criteria, see "Deduplication Comparison Parameters Form " on page 659</td></tr>
<tr><td>Name
SourceNode
SourceObject</td><td>**Note**: Select this option only if the Source Node and Source Object are stored in the NNMi database.
Each of the following values must match:
- **Name** attribute from the Incident form: General tab
- The **Source Node** attribute value from the Basics attributes listed on the Incident form
- The **Source Object** attribute value from the Basics attributes listed on the Incident form</td></tr>
<tr><td>Name
SourceNode
SourceObject
CIA</td><td>**Note**: Select this option only if the Source Node and Source Object are stored in the NNMi database.
Each of the following values must match:
- **Name** attribute from the Incident form: General tab
- The **Source Node** attribute value from the Basics attributes listed on the Incident form
- The **Source Object** attribute value from the Basics attributes listed on the Incident form
- **CIA** - Represents the Value associated with any of the following items configured as a Parameter Value using the "Deduplication Comparison Parameters Form " on page 659:
  - The **Name** attribute from the Incident form: Custom Attributes tab
  - An SNMP varbind Object ID
  - An SNMP varbind position number
If you want to use CIA as part of your comparison criteria, see "Deduplication Comparison Parameters Form " on page 659</td></tr>
</table>
</td></tr>
<tr><td>Deduplication Comparison Parameters</td><td>*Optional*. If you selected a Comparison Criteria that includes CIA, you must populate one or more rows in this table. See "Deduplication Comparison Parameters Form " on page 659.</td></tr>
</table>

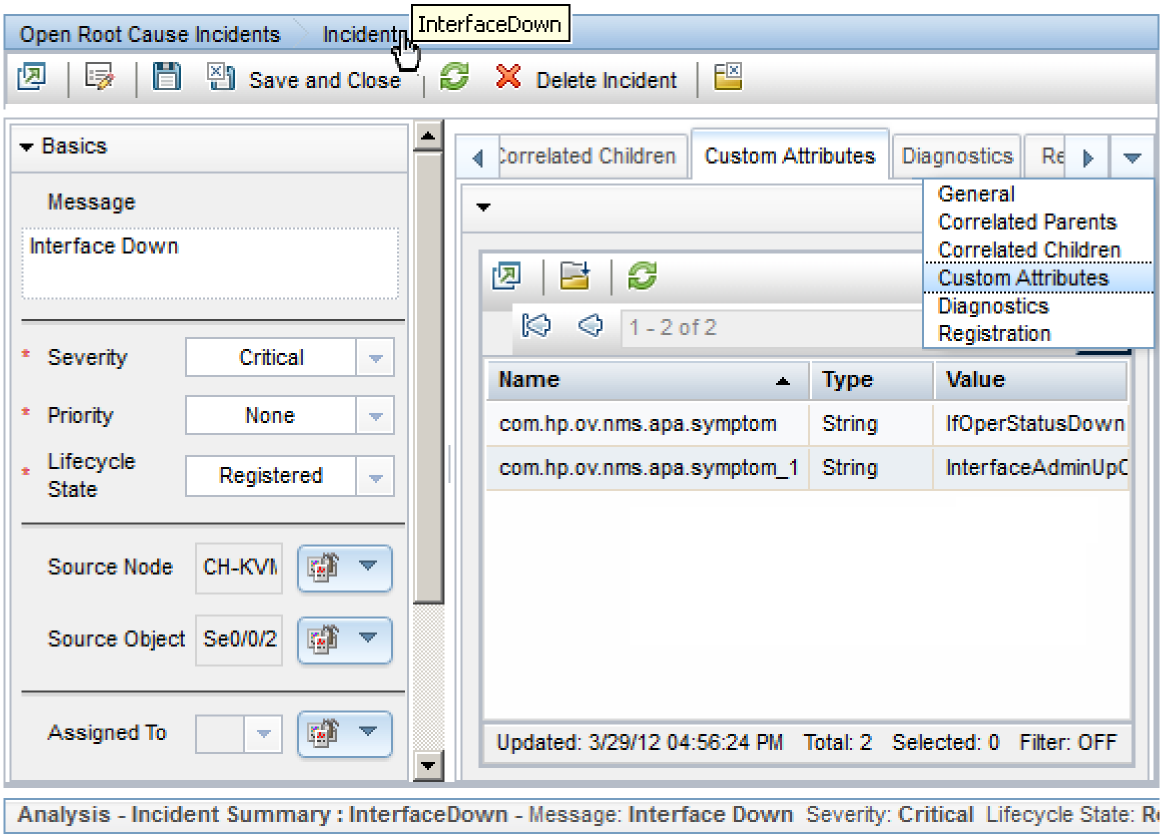
Deduplication Comparison Parameters Form (Remote NNM 6.x/7.x Events)

Comparison Parameter values enable accurate identification of duplicate incidents. Custom Incident Attributes (CIAs) are used as Comparison Parameter values. There are two categories of CIAs:

- SNMP trap varbind values (Name = the MIB varbind identifier, Type = asn_*)
- Custom attributes provided by NNMi (Name = cia.*, Type=String). See "Custom Incident Attributes Provided by NNMi (Information for Administrators)" on page 647.

The group of available CIAs depends on which incident you are configuring for this Deduplication (for example, CiscoLinkDown). To see which CIAs are available, navigate to an Incident view, select an instance of that incident-type, click the Open icon and navigate to the Custom Attributes tab. The items listed in the table are the CIAs for that particular incident-type. For example, all CiscoLinkDown incidents would have the same group of CIAs shown in the illustration below.

Note: You can also use the CIA (varbind) position number.



Message: Interface Down

* Severity	Critical
* Priority	None
* Lifecycle State	Registered
Source Node	CH-KVI
Source Object	Se0/0/2
Assigned To	

Name	Type	Value
com.hp.ov.nms.apa.symptom	String	IfOperStatusDown
com.hp.ov.nms.apa.symptom_1	String	InterfaceAdminUpC

Updated: 3/29/12 04:56:24 PM Total: 2 Selected: 0 Filter: OFF

Analysis - Incident Summary : InterfaceDown - Message: Interface Down Severity: Critical Lifecycle State: R

To specify a CIA to use in the identification criteria for duplicate incidents:

1. Navigate to the **Deduplication Comparison Params** form.
   a. From the workspace navigation panel, select the **Configuration** workspace.

b. Expand the **Incidents** folder.

c. Select **Remote NNMi 6.x/7.x Event Configurations**.

d. Do one of the following:

   - To create a new configuration, click the New icon.
   - To edit an existing configuration, select a row, click the Open icon, and continue.

e. On the form that opens, navigate to the **Deduplication** tab.

f. Locate the **Deduplication Comparison Parameters** table.

g. Do one of the following to specify which CIA:

   - To add a Custom Incident Attribute parameter specification, click the New icon.
   - To edit an existing Custom Incident Attribute parameter specification, select a row, click the Open icon, and continue.

2. In the Parameter Value field, type (or copy and paste) the exact text string from the Incident form, Custom Attribute tab, **Name** attribute value:
   - NNMi-provided CIA value (see "Custom Incident Attributes Provided by NNMi (Information for Administrators)" on page 647).
   - SNMP trap varbind identified by the Abstract Syntax Notation value (ASN.1).
3. Click **Save and Close** to save your changes and return to the previous configuration form.

Configure Rate (Time Period and Count) for a Remote NNM 6.x/7.x Event Incident

For information about each Remote NNM 6.x/7.x Events tab:

Use Rate configuration to track incident patterns *based on the number of incident reoccurrences within a specified time period*. After the count within the specified time period is reached, NNMi emits a Rate Correlation incident and continues to update the Correlation Notes with the number of occurrences within that rate.

Note: Suppression, Enrichment, and Dampening are not supported for Rate incidents.

As long as your defined criteria (Count and Hours, Minutes, Seconds) is sustained, the following information is updated in the Correlation Notes of the Rate Correlation incident:

- the actual number of occurrences of incidents for that sustained rate (Count)
- the sustained time interval (Hours, Minutes, Seconds)

For example, you can set a Rate configuration to track when a link is intermittently down at least three times in 30 minutes. NNMi shows the first occurrence of the rate incident in the incident view and uses Correlation Notes to update the number of incidents and time interval to reflect all the incremental incident occurrences and time periods. To continue the example, if the rate of three times in 30 minutes is sustained for 90 minutes, NNMi updates the Correlation Notes to specify that 9 incidents occurred in 90 minutes.

NNMi provides preconfigured Rate correlations. You can add new Rate correlations.

When you open the Incident form of the newest instance:

- On the General tab, two fields notify you that the Rate correlation is working:
  - **Correlation Nature**: Rate
  - **Count**: x
- On the **Correlated Children** tab, each incident is listed in the table.
- If a Rate Correlation Incident is dampened, note the following:
  - Rate Correlation Incidents inherit the Dampening configuration settings from its Correlated Children.
  - NNMi always retains the Parent Rate Correlation Incident, even if its Child Incidents are Closed and subsequently deleted.

    See "Dampening Incident Configurations" on page 679 for more information about Dampening an incident configuration.

To establish a rate correlation within an incident configuration:

1. Navigate to the **Rate** tab.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Expand the **Incidents** folder.
   c. Select **Remote NNM 6.x/7.x Event Configurations** .
   d. Do one of the following:
      - To create a new configuration, click the ✱ New icon.
      - To edit an existing configuration, select a row, click the Open icon, and continue.
   e. On the form that opens, locate the **Rate** tab.
2. Provide the definition for this Rate configuration (see the "Rate Configuration Definition" table).
3. *Optional*. If your Comparison Criteria includes custom incident attributes (CIA) to identify one specific incident, use the Comparison Parameter List table to define each CIA.See "Rate Comparison Parameters Form" on page 678.
4. Click **Save and Close** to save your changes and return to the previous form.

Rate Configuration Definition

Attribute	Description
Enabled	Use this attribute to temporarily disable an incident's rate settings: **Enable** □ = Temporarily disable the selected configuration. **Enable** ☑ = Enable the selected configuration. If enabled, NNMi actively tracks any reoccurrences of the designated incident within the time period you specify, and generates a Rate incident.
Count	Specify the number of reoccurrences required before your Rate Configuration starts working.

Rate Configuration Definition , continued

Attribute	Description
Hours	Used with the Minutes and Seconds attributes to specify the time duration within which the reoccurrences are measured.
Minutes	Used with the Hours and Seconds attributes to specify the time duration within which the reoccurrences are measured.
Seconds	Used with the Hours and Minutes attributes to specify the time duration within which the reoccurrences are measured.
Parent Incident	Click the icon and select Quick Find. Select **Rate Correlation** from the list.
Comparison Criteria	Specify which group of attributes must match before the incident is identified as a duplicate. The possible groups of attributes consist of the following choices. **Name** value of the Incident (from the General tab on the Incident form). **Source Node** value (from the Basics group on the Incident form). Address or name of the node for which the incident was generated. **Source Object** value (from the Basics group on the Incident form). For example, the Source Object for a LinkDown incident is **interface**. **CIA** custom incident attribute values (select from the list displayed on the Custom Attributes tab on the Incident form). If you want to use CIA as part of your comparison criteria, see "Rate Comparison Parameters Form (Remote NNM 6.x/7.x Events)" below.
Rate Comparison Parameters	*Optional*. If you selected a Comparison Criteria that includes CIA, you must populate one or more rows in this table. See "Rate Comparison Parameters Form (Remote NNM 6.x/7.x Events)" below.

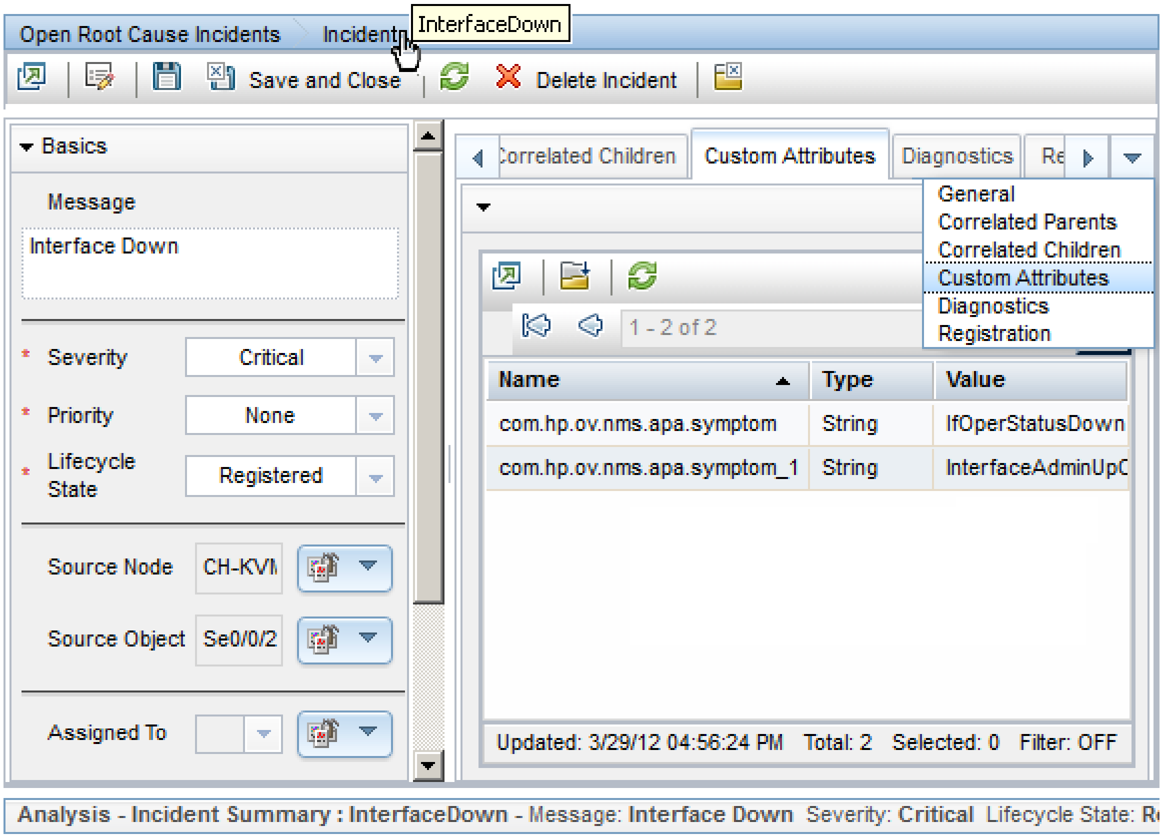
Rate Comparison Parameters Form (Remote NNM 6.x/7.x Events)

Custom Incident Attributes (CIAs) are used as parameter values. Parameter values enable accurate identification of duplicate incidents. There are two categories of CIAs:

- SNMP trap varbind values (Name = the MIB varbind identifier, Type = asn_*)
- Custom attributes provided by NNMi (Name = cia.*, Type=String). See "Custom Incident Attributes Provided by NNMi (Information for Administrators)" on page 647.

The group of available CIAs depends on which incident you are configuring for this Rate (for example, CiscoLinkDown). To see which CIAs are available, navigate to an Incident view, select an instance of that incident-type, click the Open icon and navigate to the Custom Attributes tab. The items listed in the table are the CIAs for that particular incident-type. For example, all CiscoLinkDown incidents would have the same group of CIAs shown in the illustration below.

Note: You can also use the CIA (varbind) position number.



Message: Interface Down

* Severity	Critical
* Priority	None
* Lifecycle State	Registered
Source Node	CH-KVI
Source Object	Se0/0/2
Assigned To	

Name	Type	Value
com.hp.ov.nms.apa.symptom	String	IfOperStatusDown
com.hp.ov.nms.apa.symptom_1	String	InterfaceAdminUpC

Updated: 3/29/12 04:56:24 PM Total: 2 Selected: 0 Filter: OFF

Analysis - Incident Summary : InterfaceDown - Message: Interface Down Severity: Critical Lifecycle State: R

To specify a CIA to use in the identification criteria for duplicate incidents:

1. Navigate to the **Rate Comparison Params** form.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Expand the **Incidents** folder.
   c. Select **Remote NNMi 6.x/7.x Event Configurations** .
   d. Do one of the following:
      - To create a new configuration, click the ✳ New icon.
      - To edit an existing configuration, dselect a row, click the 📂 Open icon, and continue.
   e. On the form that opens, navigate to the **Rate** tab.
   f. Locate the **Rate Comparison Parameters** table.
   g. Do one of the following to specify which CIA:
      - To add a Custom Incident Attribute parameter specification, click the ✳ New icon.
      - To edit an existing Custom Incident Attribute parameter specification,select a row, click the 📂 Open icon, and continue.
2. In the Parameter Value field, type (or copy and paste) the exact text string from the Incident form, Custom Attribute tab, **Name** attribute value:

- NNMi-provided CIA value (see "Custom Incident Attributes Provided by NNMi (Information for Administrators)" on page 647).
- SNMP trap varbind identified by the Abstract Syntax Notation value (ASN.1).

3. Click **Save and Close** to save your changes and return to the previous configuration form.

Configure Actions for a Remote NNM 6.x/7.x Event Incident

For information about each Remote NNM 6.x/7.x Events tab:

For information about each Action tab:

You can configure actions to automatically run at any point in the incident lifecycle. For example, you might want to configure an action to occur when an incident of the type you are configuring is generated (**Registered**). When an incident is generated, you might want to automatically open a trouble ticket or send email or page your network operator. After the incident is **Closed**, you might want to automatically close the trouble ticket.

> **Note:** Your actions will not be executed until you enable the Actions configuration by either clicking Enable ☑ on the Actions tab or using the **Actions** → **Enable Configuration** option.

> **Note:** If the NNMi management server is running on a Windows operating system, NNMi runs each action that you configure using the Local System account. If the NNMi management server is running on an HP-UX, Solaris or Linux operating system, NNMi runs each action that you configure using the bin user name. To change the user account associated with actions, see the "Setting the Action Server Name Parameter" section in the *HP Network Node Manager i Software Deployment Reference*.

You can configure actions for incidents generated from SNMP traps, NNM 6.x or 7.x events, and the NNMi management events. Any time an incident configuration changes, the action directory is rescanned and any Jython files are reloaded to the NNMi database. See "Lifecycle Transition Action Form (Remote NNM 6.x/7.x Events)" on the next page for more information about the actions directory.

> **Tip:** Copy any required Jython files to the NNMi actions directory before you configure an incident action. New or updated actions are loaded into NNMi only when an incident configuration is updated or created. See "Lifecycle Transition Action Form (Remote NNM 6.x/7.x Events)" on the next page for the location of the NNMi action directory.

When the action runs, output is logged to the `incidentActions.*.*.log` file. To view the contents of the Actions log, use the **Tools** → **Incident Actions Log** menu option.

See "Verify that NNMi Services are Running" on page 85 for more information about log files and where they are located.

NNMi sets the default values described in the following table.

See the "Maintaining NNMi" chapter in the *HP Network Node Manager i Software Deployment Reference* for information about changing the default values for Action Server Properties.

Action Server Properties

Property	Description	Value
numProcess	Number of actions that can be run at one time.	10
numJythonThreads	Number of threads the action server uses to run Jython scripts	10
userName	User name under which the action server runs.	bin

To configure an automatic action for an incident:

1. Navigate to the **Actions** tab.
    a. From the workspace navigation panel, select the **Configuration** workspace.
    b. Expand the **Incidents** folder
    c. Select **Remote NNM 6.x/7.x Event Configurations** .
    d. Do one of the following:
        - To create an incident configuration, click the New icon, and continue.
        - To edit an incident configuration, select a row, click the Open icon, and continue.
        - To delete an incident configuration, select a row and click the Delete icon.
    e. Select the **Actions** tab.
2. From the **Lifecycle Actions** table toolbar, do one of the following:
    - To create an Action configuration, click the New icon, and continue.
    - To edit an Action configuration, select a row, click the Open icon, and continue.
    - To delete an Action configuration, select a row and click the Delete icon.
3. In the "Lifecycle Transition Action Form (Remote NNM 6.x/7.x Events)" below, provide the required information.
4. Click **Save and Close** to save your changes and return to the previous form.

    The next time the lifecycle changes, NNMi launches the action associated with the lifecycle for the incident of that type .

Lifecycle Transition Action Form (Remote NNM 6.x/7.x Events)

For information about each Action tab:

Use this form to enter the command you want to run when an incident of the type you are configuring is at a particular Lifecycle State. For example, when an incident is generated (**Registered**), you might want to automatically open a trouble ticket or email or page your network operator.

Note: Your actions will not be executed until you enable the Actions configuration by either clicking Enable ☑ on the Actions tab or using the **Actions** → **Enable Configuration** option.

To configure an action for an incidents:

1. Navigate to the **Lifecycle Transition Actions** form:
   a. From the workspace navigation pane, select the **Configuration** workspace.
   b. Expand the **Incidents** folder.
   c. Select **Remote NNM 6.x/7.x Event Configurations**.
   d. Select the **Actions** tab.
   e. From the **Lifecycle Transition Action** table toolbar, do one of the following:
      - To create an Action configuration, click the ✳ New icon, and continue.
      - To edit an Action configuration, select a row, click the 📂 Open icon, and continue.
      - To delete an Action configuration, select a row and click the ✖ Delete icon.
2. Make your configuration choices (see table).

   Note: NNMi reloads the configuration information anytime the incident configuration is changed.
3. Click 💾 **Save and Close** to save your changes and return to the previous form.

Create Action Attributes

Attribute	Description
Lifecycle State	Select a Lifecycle State from the drop-down menu.
Command Type	If you provided a Jython command, select **Jython** from the drop-down list. If you are using an executable or bat file, select **ScriptOrExecutable** from the drop-down list.
Command	Enter one of the following: • A Jython method with the required parameters • Executable command for the current operating system with the required parameters. When entering a **Command** value, note the following: • Left or right bracket ([ ]) and backtick (` Unicode character: 0060 hex = 96 dec) characters are not permitted in the **Command** attribute. If you need these characters in your shell script, place them in a shell script file and reference that file from the **Command** attribute. • **Windows only**: Shell commands are not permitted in the **Command** attribute. To use shell commands, place them in a shell script file and reference that file from the **Command** attribute. • Use absolute paths to executables instead of relying on the PATH variable as it might not be set correctly. • Verify that you do not have two Jython methods with the same name. Otherwise,

Create Action Attributes, continued

Attribute	Description

	NNMi is not able to tell which is the correct method to load.
	• You can use the same Jython method for more than one incident configuration.
	• Jython (.py) files must reside in the following directory:
	Note: All the functions defined in the Jython files that reside in this directory are also accessible by NNMi.The files are also executed by NNMi on startup.
	Windows:
	`%NnmDataDir%\shared\nnm\actions`
	UNIX:
	`/var/opt/OV/shared/nnm/actions`
	• When using executable files, specify the absolute path to the executable command or make sure the directory in which the executable file resides is in your PATH environment variable.
	• NNMi provides a set of parameters that can pass attribute values from an incident configuration. See "Valid Parameters for Configuring Incident Actions (Management Events)" on page 1216 for more information.

Configure a Payload Filter for an Action (Remote NNM 6.x/7.x Events)

The Payload Filter Editor enables you to create expressions that further refine the filters used to select the incidents that cause the configured action to run. Make sure to design any complex Payload Filters offline as a Boolean expression first. This method can help to minimize errors when entering your expressions using the Payload Filter editor.

To create a Payload Filter expression:

1. Navigate to the **Remote NNM 6.x/7.x Event Configuration** form:
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Expand the **Incidents** folder.
   c. Select **Remote NNM 6.x/7.x Event Configurations**.
   d. Do one of the following:
      i. To create an incident configuration, click the New icon, and continue.
      ii. To edit an incident configuration, select a row, click the Open icon, and continue.
      iii. To delete an incident configuration, select a row and click the Delete icon.

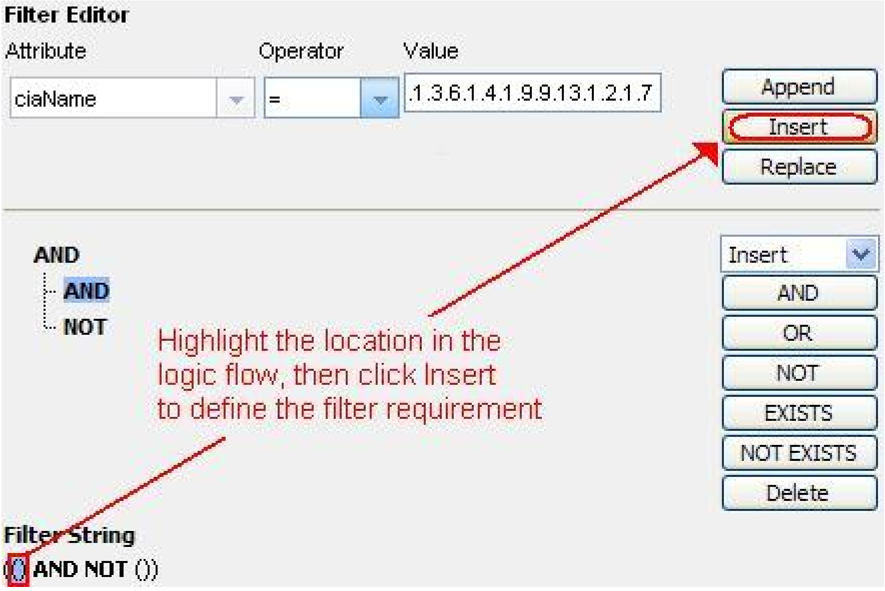
2. Select the **Actions** tab.
3. Do one of the following:
   a. To create a new configuration, click the * New icon.
   b. To edit an existing configuration, select a row, click the Open icon, and continue.
4. Select the **Payload Filter** tab.
5. Define your Payload Filter (see table). Also see "Guidelines for Creating a Payload Filter".
   a. Plan out the logic needed for your Filter String.
   b. Use the buttons on the bottom half of the Additional Filters Editor to establish the logic structure.

      For example, to establish the following structure, select **Insert**, then click **AND**, then **NOT**, and then **AND** a second time:

      ```
      (( ) AND NOT ( ))
      ```

   c. Now place your cursor in a location within the displayed Filter String, and use the top half of the filter editor to define the parameters of the highlighted filter requirement.

      For example, select a set of parentheses and use the Insert button to specify the filter requirement within those parentheses:



Attribute: ciaName

Operator: =

Value: .1.3.6.1.4.1.9.9.13.1.2.1.7

Highlight the location in the logic flow, then click Insert to define the filter requirement

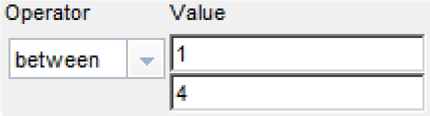
Filter String: (() AND NOT ())

6. Click **Save and Close**.
7. Click **Save and Close** to save your changes and return to the previous form.

Payload Filter Editor Components

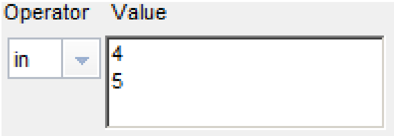
Attribute	Description
Attribute	The attribute name on which NNMi searches. Filterable attributes include the following:

Payload Filter Editor Components, continued

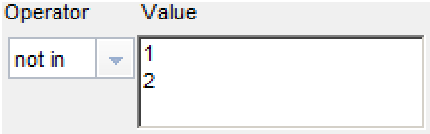
Attribute	Description
	- ciaName - ciaValue
Operator	Valid operators are described below. - **=** Finds all values equal to the value specified. Click here for an example. Example: `ciaName = .1.3.6.1.4.1.9.9.13.1.2.1.7` matches any incident that contains a varbind with the name value **.1.3.6.1.4.1.9.9.13.1.2.1.7**. - **!=** Finds all values not equal to the value specified. Click here for an example. Example: `ciaName != .1.3.6.1.4.1.9.9.13.1.2.1.7` matches any incident that contains a varbind with the name value other than **1.3.6.1.4.1.9.9.13.1.2.1.7**. - **<** Finds all values less than the value specified. Click here for an example. Example: `ciaValue < 6` matches any incident that contains a varbind value less than **6**. - **<=** Finds all values less than or equal to the value specified. Click here for an example. Example: `ciaValue <= 6` matches any incident that contains a varbind value less than or equal to **6**. - **>** Finds all values greater than the value specified. Click here for an example. Example: `ciaValue > 4` matches any incident that contains a varbind value greater than **4**. - **>=** Finds all values greater than or equal to the value specified. Click here for an example. Example: `ciaValue >= 4` matches any incident that contains a varbind with values greater than or equal to **4**. - **between** Finds all traps or events that include a varbind value equal to and between the two values specified. Click here for an example. Example: `ciaValue between`  Operator Value between 1 4 matches any incident that contains a varbind value equal to or greater than **1** and equal to or less than **4** . **Note**: As shown in the example, each value must be entered on a separate line. - **in** Finds any match to at least one value in a list of values. Click here for an example.

Payload Filter Editor Components, continued

Attribute	Description

	Example:
	`ciaValue in`
	
	Operator: in
	Value: 4
	5
	matches any incident that contains a varbind value of either **4** or **5**.
	Note: As shown in the example, each value must be entered on a separate line.
	NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example (**4, 5**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line.
	• **is not null** Finds all non-blank values. Click here for an example.
	Example: `ciaValue is not null` matches any incident with a varbind value.
	• **is null** Finds all blank values. Click here for an example.
	Example: `ciaValue is null` matches any incident with no varbind values.
	• **like** Finds matches using wildcard characters. Click here for more information about using wildcard characters.
	The period asterisk (.*) characters mean *any number of characters of any type at this location*.
	The period (.) character means *any single character of any type at this location*.
	Note: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below.
	Examples:
	`ciaName like  \Q.1.3.6.1.4.1.9.9\E.*` finds all traps or events that contain varbind names that begin with **.1.3.6.1.4.1.9.9** and (optionally) end with any number of characters.
	`ciaValue like .*Chicago.*` finds all traps or events that contain a varbind value that includes the string **Chicago**.
	• **not between** Finds all values except those between the two values specified. Click here for an example.
	Example: `ciaValue not between 5 8` matches an incident that contains a varbind with the values less than **5** or greater than **8**.
	• **not in** Finds all values except those included in the list of values. Click here for an example.
	Example:
	`ciaValue not in`

Payload Filter Editor Components, continued

Attribute	Description
	 Operator: not in Value: 1 2 matches any incident that contains a varbind with values other than **1** and **2**. **Note**: As shown in the example, each value must be entered on a separate line. NNMi displays the list of attributes using comma-separated values enclosed in parentheses, for example, (**1, 2**). However, the comma-separated list is used only for display purposes. The actual delimiter is the new line. • **not like** Finds all that do not have the values specified (using wildcard strings). Click here for an example. The period asterisk (.*) characters mean *any number of characters of any type at this location*. The period (.) character means *any single character of any type at this location*. **Note**: To include literal string values in the Value attribute, enclose the string value in \Q<*literal_value*>\E as shown in the Examples listed below. Example: `ciaName not like \Q.1.3.6.1.4.1.9.9\E.*` matches any incident that contains a varbind name value that does not begin with **.1.3.6.1.4.1.9.9**. `ciaValue not like .*Chicago.*` finds all traps or events that do not contain a varbind value that includes the string **Chicago**.
Value	The value for which you want NNMi to search. Note the following: • The values you enter are case sensitive. • NNMi displays a variable number of value fields depending on the Operator selected. For example, the `between` Operator causes two value fields to be displayed. • The `between, in` and `not in` operators require that each value be entered on a separate line.

Additional Filters Editor Buttons

Button	Description
Append	Appends the current expression (Attribute, Operator,and Value) to the selected expression already included in the Filter String.
Insert	Inserts the current expression (Attribute, Operator,and Value) in front of the cursor location within the Filter String.

Additional Filters Editor Buttons, continued

Button	Description
Replace	Replaces the selected expression with the expression displayed in the Attribute, Operator, and Value fields.
AND	Appends, inserts, or replaces the AND Boolean Operator in the selected cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
OR	Appends, Inserts, or replaces the OR Boolean Operator in the current cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
NOT	Can be used in any part of the Filter String to specify that NNMi should exclude interfaces with values that pass the expression that immediately follows the `NOT`. For example, when evaluating the following expression, NNMi includes interfaces with (interface description) `ifDesc` containing **VLAN**, and excludes any Interfaces that have **VLAN10** for the (interface name) `ifName` value: `(ifDesc like VLAN AND NOT (ifName=VLAN10))` **Note**: View the expression displayed under Filter String to see the logic of the expression as it is created .
EXISTS	Used for filters that include Capabilities or Custom Attribute names and values in the Filer String. Indicates that you want NNMi to consider interfaces that do not have any Capabilities or Custom Attributes when evaluating the Filter String. **Note**: If you include Capabilities or Custom Attribute names and values in the Filer String, but do not use **EXISTS** or **NOT EXISTS**, NNMi excludes from its search interfaces that do not include Capabilities or Custom Attributes. For example, when evaluating the following Filter String, NNMi includes interfaces with (interface description) `ifDesc` containing **VLAN**, as well as any Interfaces Custom Attribute Role value is **LAN Connection to Oracle Server**: `(ifDesc like VLAN OR EXISTS((customAttrName=Role AND customAttrValue=LAN Connection to Oracle Server)))` **Note**: View the expression displayed under Filter String to see the logic of the expression as it is created.
NOT EXISTS	Used for filters that include Capabilities or Custom Attribute names and values in the Filer String. Indicates that you want NNMi to consider interfaces that do not have any Capabilities or Custom Attributes when evaluating the Filter String, but exclude the interfaces that match the expression that follows the **NOT EXISTS**.

Additional Filters Editor Buttons, continued

Button	Description
	Note: If you include Capabilities or Custom Attribute names and values in the Filer String, but do not use **EXISTS** or **NOT EXISTS**, NNMi excludes from its search interfaces that do not include Capabilities or Custom Attributes. For example, when evaluating the following expression, NNMi includes interfaces with (interface description) `ifDesc` containing **VLAN**, and excludes any Interfaces that have the Custom Attribute Role and that Role value is **LAN Connection to Oracle Server**: `(ifDesc like VLAN OR NOT EXISTS((customAttrName=Role AND customAttrValue=LAN Connection to Oracle Server)))` **Note**: View the expression displayed under Filter String to see the logic of the expression as it is created.
Delete	Deletes the selected expression. **Note**: If the Boolean Operator is selected, the Additional Filters Editor deletes all expressions associated with the Boolean Operator.

Valid Parameters for Configuring Incident Actions (Remote NNM 6.x/7.x Events)

When configuring incident actions, consider using incident information as part of the action. NNMi provides the following parameter values. Use these parameters as variables in your Jython methods or executable files.

Tip: See the Using the Incident Form for more information about the parameter values.

Note: NNMi stores varbind values as custom incident attributes (CIAs).

Tip: If a value is not stored for a parameter, it is returned as “null”.

See "Lifecycle Transition Action Form" on page 748 for more information about configuring incident actions.

Valid Parameters Visible From an Incident's Form

Parameter Value	Description
$category, $cat	Value of the Category attribute in the Incident form.
$count, $cnt	Value representing the number of Custom Incident Attributes that appear in the Incident form.
$family, $fam	Value from the Family attribute in the Incident form.
$firstOccurrenceTime, $fot	Value from the First Occurrence Time attribute in the incident form.
$lastOccurrenceTime, $lot	Value from the Last Occurrence Time attribute in the incident form.

Valid Parameters Visible From an Incident's Form, continued

Parameter Value	Description
$lifecycleState, $lcs	Value from the Lifecycle State attribute in the Incident form.
$name	Value of the Name attribute from the incident configuration.
$nature, $nat	Value from the Nature attribute in the Incident form.
$origin, $ori	Value from the Origin attribute in the Incident form.
$originOccurrenceTime, $oot	Value from the Origin Occurrence Time attribute in the incident form.
$priority, $pri	Value from the Priority attribute in the Incident form.
$severity, $sev	Value of the Severity attribute of the Incident form.

Valid Parameters Visible from a Node Form

Parameter Value	Description
$managementAddress, $mga	Value from the Management Address attribute of the incident's source Node's form or SNMP Agent form.
$otherSideOfConnectionManagementAddress, $oma	If the incident's Source Node is part of a Layer 2 Connection, this attribute is the value of the Management Address of a node on the other side of the Layer 2 Connection.
$sourceNodeLongName, $sln	The fully-qualified DNS name as displayed in the Hostname attribute of the incident's source Node's form.
$sourceNodeName, $snn	Value from the Name attribute of the incident's source Node's form.
$sysContact, $sct	Value from the System Contact attribute of the incident's source Node form: General tab.
$sysLocation, $slc	Value from the System Location attribute of the incident's source Node form: General tab.

Valid Parameters Visible from an Interface Form

Parameter Value	Description
$ifAlias, $ifa	Value from the IfAlias attribute for the interface that is the incident's source object.
$ifConfigDupSetting,	Configured Duplex Setting on the port associated with the interface that

Valid Parameters Visible from an Interface Form , continued

$icd	is the incident's source object.
$ifDesc, $idc	Value from the ifDesc attribute for the interface that is the incident's source object.
$ifIndex, $idx	Value from the ifIndex attribute for the interface that is the incident's source object.
$ifIpAddr, $iia	IP Address values associated with the interface that is the incident's source object. If multiple IPaddresses are associated with the interface, this parameter returns a comma-separated list.
$ifName, $ifn	Value from the ifName attribute for the interface that is the incident's source object.
$ifPhysAddr, $ipa	Value from the Physical Address attribute for the interface that is the incident's source object.
$ifSpeed, $isp	Value from the ifSpeed attribute for the interface that is the incident's souce object.
$ifType, $itp	Value from the ifType attribute for the interface that is the incident's souce object.

Valid Parameters Visible from a Layer 2 Connection Form

Parameter Value	Description
$otherSideOfConnectionConfigDupSetting, $ocd	If the incident's source Node is part of a Layer 2 Connection, this parameter contains the Configured Duplex Setting on the port associated with the interface on the other side of the connection.
$otherSideOfConnectionIfAlias, $oia	If the incident's Source Node is part of a Layer 2 Connection, this parameter is the value of the ifAlias of one of the interfaces on the other side of the Layer 2 Connection.
$otherSideOfConnectionIfDesc, $odc	If the incident's Source Node is part of a Layer 2 Connection, this parameter contains the ifDescr attribute value for the interface on the other side of the Layer 2 Connection.
$otherSideOfConnectionIfIndex, $odx	If the incident's Source Node is part of a Layer 2 Connection, this parameter contains the ifIndex attribute value for the interface on the other side of the connection.
$otherSideOfConnectionIfName, $ofn	If the incident's Source Node is part of a Layer 2 Connection, this parameter contains the ifName attribute value for the interface on the other side of the connection.

Valid Parameters Visible from a VLAN Form

Parameter Value	Description
$impVlanIds, $ivi	Value from the VLAN Id attribute associated with the interface that is the incident's source object. To access this information from an interface form, navigate to the VLAN Port tab and open the form for the VLAN of interest. If the interface is part of more than one VLAN, this parameter returns a comma-separated list.
$impVlanNames, $ivn	Value from the Global VLAN Name attribute associated with the interface that is the incident's source object. To access this information from a Node form or Interface form, navigate to the VLAN Ports tab. If the node or interface is part of more than one VLAN, this parameter returns a comma-separated list.

Valid Parameters Not Visible From a Form

Parameter Value	Description
$id	Unique Object Identifier attribute value for the incident (unique across the entire NNMi Database).
$firstOccurrenceTimeMs, $fms	Value from the First Occurrence Time attribute in the incident form, converted to millseconds (measured since January 1, 1970, 00:00:00 GMT - Greenwich Mean Time).
$lastOccurrenceTimeMs, $lms	Value from the Last Occurrence Time attribute in the incident form, converted to millseconds (measured since January 1, 1970, 00:00:00 GMT - Greenwich Mean Time).
$messageFormat, $msg	*Valid for Incident actions only*. Message text displayed for an incident when this parameter is included as an argument to an incident action.
$oid	Value of the unique object identifier (oid) for the incident configuration that originated from either an SNMP trap, Management Event, or Remote NNM 6.x or 7.x event.
$otherSideOfConnection, $osc	If the incident's Source Node is part of a Layer 2 Connection, this attribute is the following combination of values for the node and one of its interfaces on the other side of the Layer 2 Connection: The fully-qualified DNS name of the node appended with the interface Name in the following format: <*fully-qualified DNS name*>[*interface_name*]
$originOccurrenceTimeMs, $oms	Value from the Origin Occurrence Time attribute in the incident form, converted to millseconds (measured since January 1, 1970, 00:00:00 GMT - Greenwich Mean Time).

Valid Parameters Not Visible From a Form, continued

Parameter Value	Description
$sourceNodeUuid, $snu	Universally Unique Object Identifier attribute value of the source node object for the incident (unique across all databases). This identifier distinguishes the source node object instance from all other node objects.
$sourceObjectClass, $soc	Value of the object class for the object you want to include. Use this parameter to request more details of a class of objects through a web service. Examples of object classes include: `com.hp.ov.nms.model.core.Interface` and `com.hp.ov.nms.model.snmp.SnmpAgent`.
$sourceObjectName, $son	Value from the Name attribute of the source object. For example, an interface object is named according to the MIB ifName. Each ifName varies according to the vendor's conventions. Using the name `4/1` as an example, `4` represents `the board number and 1` represents the port number.
$sourceObjectUuid, $sou	Universally Unique Object Identifier attribute value of the source object for the incident (unique across all databases). This identifier distinguishes the source object instance from all other similar object instances..
$uuid	Universally Unique Object Identifier attribute value of the incident (unique across all databases). This identifier distinguishes the incident object instance from all other incident objects.

Valid Parameters Established in Custom Incident Attributes

Parameter Value	Description
$<position_ number>	Value of the custom incident attribute (CIA) position number for any CIA that originated from a varbind or was added by NNMi. For example, to indicate you want to use the varbind in position 1, enter: `$1` NNMi stores varbind values as Custom Incident Attributes. If you know the varbind position number, use this parameter.
$<CIA_ name>	Value of the name that is used for the custom incident attribute. For example, `$mycompany.mycia.` NNMi provides CIA values for configuring Management Events. See Custom Incident Attributes Provided by NNMi for more information about custom incident attributes.
$<CIA_ oid>	Value of the object identifier for any custom incident attribute that originated as a varbind. For example, `$.1.3.6.1.6.3.1.1.5.1`. Use this parameter when you are not certain of a custom incident attribute (varbind) position number.
$*	Used to indicate you want all of the custom incident attribute values originating as varbinds, to be passed to the action configuration. Each varbind is returned in the

Valid Parameters Established in Custom Incident Attributes, continued

Parameter Value	Description
	following format: `$<CIA_name>:<CIA_value>` in which the custom incident attribute name appears followed by the custom incident attribute value.

The function described in the following table replaces the specified numeric value with the associated text value stored in the CIA.

Note: The associated MIB must have been loaded using the nnmloadmib.ovpl command.

Functions to Generate Values Within Incident Messages

Function	Description
$text ($<position_number>)	The <*position_number*> argument specifies the numeric value of the custom incident attribute (CIA) position number for any CIA that originated from a varbind or was added by NNMi. For example, to indicate you want to use the varbind in position 1, enter: `$1`. After the function runs, NNMi replaces the numeric value with the text value stored in the CIA. **Note**: If a text value is not available, NNMi returns the numeric value.
$text ($<CIA_oid>)	The <CIA_oid> argument specifies the object identifier for any custom incident attribute that originated as a varbind. For example, `$.1.3.6.1.6.3.1.1.5.1`. Use this argument to the $text function when you are not certain of a custom incident attribute (varbind) position number. After the function runs, NNMi replaces the numeric value with the text value stored in the CIA. **Note**: If a text value is not available, NNMi returns the numeric value.

Troubleshoot Incident Configurations

The NNMi **Actions** menu enables you to open an Incident Configuration from either an incident or an incident view. This feature is useful when you are monitoring incoming incidents to determine whether incidents are generated as expected. After you make any required changes, you can easily verify your changes the next time the incident occurs.

Note: Your User Account must be assigned to the **NNMi Administrators** User Group to use these actions.

Tip: You can right-click any object in a table or map view to access the **Actions** menu.

To open an Incident Configuration form from an incident view:

1. Navigate to the incident view of interest. (For example, select the **Incident Browsing** workspace, **Root Cause Incidents** view.)
2. In the table view, press CTRL-Click and select each row representing an incident of interest.
3. In the main toolbar, select **Actions** → **Open Incident Configuration Form**.

   NNMi opens one Incident Configuration form for each type of incident selected.

To open an Incident Configuration form from an Incident form:

1. Navigate to the incident view of interest. (For example, select the **Incident Browsing** workspace, **Root Cause Incidents** view.)
2. In the table view, press CTRL-Click and select each row representing the configuration you want to edit.
3. In the main toolbar, select **Actions** → **Open Incident Configuration Form**.

   NNMi opens the Incident Configuration form for the current incident.

Note: Any configuration changes you make to an incident apply only to future incidents.
The NNMi **Actions**→ **Incident Configuration Report** menu also enables you to view configuration reports for the following kinds of configurations for an incident:

- Action Results
- Dampen Results
- Enrichments
- Global Manager Forwarding (*NNMi Advanced -Global Network Management*) Available on Regional Managers.
- Suppression Results

See "View an Incident Configuration Report" below for more information.

View an Incident Configuration Report

The NNMi **Actions** menu enables you to view a report of the following incident configurations:

- Action Results
- Dampen Results
- Enrichment
- Global Manager Forwarding (*NNMi Advanced - Global Network Management feature*)
- Suppression Results

Note: Your User Account must be assigned to the **NNMi Administrators** User Group to use these actions.

Tip: You can right-click any object in a table or map view to access the **Actions** menu.

Viewing an incident configuration report helps you determine the following:

- (*NNMi Advanced - Global Network Management feature*). On a Regional Manager, reports whether NNMi forwards occurrences of the selected incident configuration to Global Managers.
- The configuration settings (Interface, Node, or Default) NNMi is using for a selected incident.
- Whether the selected configuration (Suppression, Enrichment, or Dampening) is enabled.
- Whether NNMi found any matches for a Payload Filter for the selected configuration (Suppression, Enrichment, or Dampening).

These reports are useful when you want to change an incident configuration and need to determine which settings have been configured, and therefore which settings you might want to change, for the incident.

To view a configuration report for the selected incident:

1. Select the incident for which you want to view a configuration report.
2. Select **Actions→ Incident Configuration Reports.**
3. Select one of the following menu options to indicate the type of configuration report you want to view
   - **Action Results**
   - **Dampen Results**
   - **Report Enrichments**
   - **Global Manager Forwarding** (*NNMi Advanced*)
   - **Suppression Results**

See the Incident Configuration Actions table for a description of each incident configuration report.

Incident Configuration Actions

Action Menu Option	Information Displayed
Action Results	• If the Source Object is an interface, the Interface Group, if any, to which the Source Object belongs. If NNMi reports a matching Interface Group, this indicates it is using the Interface configuration settings for the selected incident. • The Node Group, if any, to which the Source Node belongs. If NNMi reports a matching Node Group, this indicates it is using the Node configuration settings for the selected incident. **Note**: If no matches for an Interface or Node Group are found, this indicates NNMi is using the Default configuration settings for the selected incident. • Whether the Action Configuration is enabled. • The action to be executed. • The Payload Filter, if configured for the incident and whether NNMi found any matches for the Payload Filter.

Incident Configuration Actions, continued

Action Menu Option	Information Displayed
Dampen Results	• If the Source Object is an interface, the Interface Group, if any, to which the Source Object belongs. If NNMi reports a matching Interface Group, this indicates it is using the Interface configuration settings for the selected incident. • The Node Group, if any, to which the Source Node belongs. If NNMi reports a matching Node Group, this indicates it is using the Node configuration settings for the selected incident. **Note**: If no matches for an Interface or Node Group are found, this indicates NNMi is using the Default configuration settings for the selected incident. • Whether the Dampening configuration is enabled. • The Dampen Interval that is set. • The Payload Filter, if configured for the incident and whether NNMi found any matches for the Payload Filter.
Report Enrichments	• If the Source Object is an interface, the Interface Group, if any, to which the Source Object belongs. If NNMi reports a matching Interface Group, this indicates it is using the Interface configuration settings for the selected incident. • The Node Group, if any, to which the Source Node belongs. If NNMi reports a matching Node Group, this indicates it is using the Node configuration settings for the selected incident. **Note**: If no matches for an Interface or Node Group are found, this indicates NNMi is using the Default configuration settings for the selected incident. • Whether the Enrichment configuration is enabled. • The Payload Filter, if configured for the incident and whether NNMi found any matches for the Payload Filter. • The current Severity, Priority, Message Format, and Custom Incident Attributes configuration settings for the incident.
Global Manager Forwarding	(*NNMi Advanced - Global Network Management feature*). Displays the following for each selected incident: • Whether the incident is an SNMP Trap, Remote NNM 6.x/7.x Management Event, or Management Event Configuration. • The name of the incident configuration. • Whether occurrences of the selected incident configuration will be forwarded to Global Managers in your network environment.

Incident Configuration Actions, continued

Action Menu Option	Information Displayed
	• The Payload Filter, if configured for the incident and whether NNMi found any matches for the Payload Filter.
Suppression Results	• If the Source Object is an interface, the Interface Group, if any, to which the Source Object belongs. If NNMi reports a matching Interface Group, this indicates it is using the Interface configuration settings for the selected incident. • The Node Group, if any, to which the Source Node belongs. If NNMi reports a matching Node Group, this indicates it is using the Node configuration settings for the selected incident. **Note**: If no matches for an Interface or Node Group are found, this indicates NNMi is using the Default configuration settings for the selected incident. • Whether the Suppress Configuration is enabled. • The Payload Filter, if configured for the incident, and whether NNMi found any matches for the Payload Filter.

Related Topics

"Troubleshoot Incident Configurations" on page 1371

Chapter 15

Configure Trap Forwarding

NNMi enables you to configure SNMP trap forwarding using the Trap Forwarding option under the Incidents folder of the 🔧 Configuration workspace. This feature is useful when you want to forward traps to a specified destination. For example, you might want to forward certain kinds of traps to one server and forward another set of traps to a different server so they can be managed separately.

When configuring SNMP trap forwarding you perform the following tasks:

- "Configure NNMi SNMPv3 Security Settings for Trap Forwarding and Inform-Requests" below
- "Configure Trap Forwarding Filters" on page 1378
- "Configure Trap Forwarding Destinations" on page 1381

Note: See "Manage Incoming SNMP Traps" on page 769 for information about the criteria NNMi uses to determine when to receive or discard traps.

(*NNMi Advanced - Global Network Management feature*) If you want to forward specific SNMP traps from your NNMi management server (a Regional Manager) to all Global Managers in your Global Network management environment, see "Configure Forward to Global Manager Settings for an SNMP Trap Incident (NNMi Advanced)" on page 927

Configure NNMi SNMPv3 Security Settings for Trap Forwarding and Inform-Requests

Note: If your network environment uses SNMPv2c or SNMPv1 and does not use SNMPv3, skip this task.

If your network environment uses SNMPv3, specify which user-based security model (USM) settings the NNMi management server uses when NNMi acts as an authoritative entity in the following situations:

- Forwarding SNMPv3 traps to other devices in your network environment
- Sending responses to SNMPv3 Inform-Requests

The settings in this form grant permission for NNMi to communicate with the SNMPv3 agent. The SNMPv3 engine identifier and the user-based security settings are required for successful authentication in SNMPv3 protocol. Devices that are sending SNMPv3 informs to NNMi must use these settings.

To configure the NNMi management server as an authoritative entity for SNMPv3:

1. Navigate to the **Trap Forwarding Configuration** form.
   a. From the workspace navigation panel, select the 🔧 **Configuration** workspace.
   b. Expand the **Incidents** folder.

   c. Expand the **Trap Server** folder.

   d. Select **Trap Forwarding Configuration.**

2. Navigate to the **NNMi SNMPv3 Trap Forwarding Security Settings** group.

3. NNMi displays the ID of the engine assigned to the SNMPv3 agent that NNMi uses when forwarding or sending data to other SNMPv3 agents. See the attribute value for Engine Id.

   Caution: Devices that are sending SNMPv3 informs to NNMi must use these settings.

4. Provide the USM information that NNMi uses for authentication and privacy when using SNMPv3 protocol for forwarding traps or receiving Inform-Requests from other devices in your network environment (see table).

5. Click **Save and Close** to save your changes.

SNMPv3 Engine Assigned to NNMi management server

Attribute	Description
Engine Id	Remote devices must request this SNMPv3 engine ID when sending informs to NNMi. If the SNMPv3 agent sending data to NNMi does not know the correct engine ID, the inform is rejected.

SNMPv3 Settings of the NNMi management server's User-Based Security Model (USM)

Attribute	Description
User Name	The SNMPv3 User Name is the text string used for SNMPv3 requests in your network environment.
Authentication Protocol	The SNMPv3 authentication protocol. Determines whether authentication is required and indicates the type of authentication protocol used. NNMi supports the following protocols: • **HMAC**[1]-**MD5**[2]-96 authentication protocol • **HMAC**[3]-**SHA**[4]-1 authentication protocol
Authentication Passphrase	The SNMPv3 USM authentication passphrase used by the NNMi management server. If required for authentication, provide the appropriate authentication passphrase for the authentication protocol. The length limitations of the authentication passphrase depend on the authentication protocol.
Privacy Protocol	Specify the SNMPv3 USM privacy protocol used by the NNMi management server.

[1] Hash-based Message Authentication Code
[2] Message-Digest algorithm 5
[3] Hash-based Message Authentication Code
[4] Secure Hash Algorithm

SNMPv3 Settings of the NNMi management server's User-Based Security Model (USM), continued

Attribute	Description
	The SNMPv3 USM privacy protocol determines whether encryption is required and indicates the type of privacy protocol used. NNMi supports the following privacy protocols: • **DES**[1]-**CBC**[2] Symmetric Encryption Protocol • Triple**DES**[3] - Triple Data Encryption Algorithm • **AES**[4]128 - Advanced Encryption Standard 128 Protocol • **AES**[5]192 - Advanced Encryption Standard 192 Protocol • **AES**[6]256 - Advanced Encryption Standard 256 Protocol **Note**: Leaving this attribute empty means SNMP Minimum Security Level = *No Privacy* for this SNMPv3 configuration.
Privacy Passphrase	Specify the SNMPv3 USM privacy passphrase used by the NNMi management server. If required for privacy, provide the appropriate encryption passphrase for use with the privacy protocol. The length limitations of the privacy passphrase depend on the privacy protocol.

Registration Attributes

Attribute	Description
Last Modified	Date and time the Trap Forwarding Configuration was last modified.

Configure Trap Forwarding Filters

Pre-requisite: Make sure you have used the NNMi nnmincidentcfg.ovpl command line utility to automatically create or update the incident configurations for the SNMP traps you want to include. See "Load SNMP Trap Incident Configurations" on page 771 for more information.

Use the Trap Forwarding Configuration: Trap Forwarding Filters tab to configure a filter expression to specify the SNMP trap Object Identifier (OID) pattern you want to use to determine which SNMP traps NNMi forwards. The traps which pass the filter you specify can then be forwarded to a specified destination using the Trap Forwarding Destinations tab. See "Configure Trap Forwarding Destinations" on page 1381 for more information.

[1] Data Encryption Standard
[2] Cipher Block Chaining
[3] Data Encryption Standard
[4] Advanced Encryption Standard
[5] Advanced Encryption Standard
[6] Advanced Encryption Standard

Note: See "Manage Incoming SNMP Traps" on page 769 for information about the criteria NNMi uses to determine when to receive or discard traps.

To configure Trap Forwarding Filters:

1. Navigate to the **Trap Forwarding Configuration** form.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Expand the **Incidents** folder.
   c. Expand the **Trap Server** folder.
   d. Select **Trap Forwarding Configuration.**
2. Select the **Trap Forwarding Filters** tab.
3. Do one of the following:
   - To create an SNMP Trap Forwarding Filter configuration, click the New icon, and continue.
   - To edit an SNMP Trap Forwarding Filter configuration, click the Open icon in the row representing the configuration you want to edit, and continue.
   - To delete an SNMP Trap Forwarding Filters configuration, click the Delete icon.
4. In the "Trap Forwarding Filter Form" below provide the required information.
5. Click **Save and Close** to save your changes and return to the **Trap Forwarding Configuration** form.

The next time that a trap of this type arrives, NNMi uses the filter you specify to determine whether to forward the trap to a specified destination.

Trap Forwarding Filter Form

Pre-requisite: Make sure you have used the NNMi nnmincidentcfg.ovpl command line utility to automatically create or update the incident configurations for the SNMP traps you want to include. See "Load SNMP Trap Incident Configurations" on page 771 for more information.

The Trap Forwarding Filters Form enables you to specify the SNMP trap Object Identifier (OID) pattern you want to use to determine which SNMP traps NNMi forwards. The traps which pass the filter you specify can then be forwarded to a specified destination using the Trap Forwarding Destinations tab. See "Configure Trap Forwarding Destinations" on page 1381 for more information.

Note: See "Manage Incoming SNMP Traps" on page 769 for information about the criteria NNMi uses to determine when to receive or discard traps.

To configure Trap Forwarding Filters:

1. Navigate to the **Trap Forwarding Configuration** form:
   a. From the workspace navigation pane, select the **Configuration** workspace.
   b. Expand the **Incidents** folder.
   c. Expand the **Trap Server** folder.

   d. Select **Trap Forwarding Configuration**.

2. Select the **Trap Forwarding Filters** tab.
3. Do one of the following:
   - To add an SNMP Trap Forwarding Filter configuration, click the New icon, and continue.
   - To edit an SNMP Trap Forwarding Filter configuration, click the Open icon in the row representing the configuration you want to edit, and continue.
   - To delete an SNMP Trap Forwarding Filter configuration, click the Delete icon.
4. Make your configuration choices (see table).
5. Click **Save and Close** to save your changes and return to the **Trap Forwarding Configuration** form.

SNMP Trap Forwarding Filters Configuration

Attribute	Description
Filter Name	Enter the name you want to use for this SNMP Trap Forwarding Filter configuration. Alpha-numeric and special characters (~ ! @ # $ % ^ & * ( ) _+ -) are permitted. No spaces are permitted.
"Filter Form" below	Access the Filter Expressions tab to access the Filter form and specify the valid SNMP Object Identifier (OID) pattern to be used for the SNMP trap filter.

Filter Form

Pre-requisite: Make sure you have used the NNMi nnmincidentcfg.ovpl command line utility to automatically create or update the incident configurations for the SNMP traps you want to include. See "Load SNMP Trap Incident Configurations" on page 771 for more information.

The Filter Form enables you to specify the SNMP trap Object Identifier (OID) pattern you want to use to filter incoming SNMP traps. The traps which pass the filter you specify can then be forwarded to a specified destination using the Trap Forwarding Destinations tab. See "Configure Trap Forwarding Destinations" on the next page for more information.

Note: See "Manage Incoming SNMP Traps" on page 769 for information about the criteria NNMi uses to determine when to receive or discard traps.

To configure a Trap Forwarding Filter:

1. Navigate to the **Trap Forwarding Filter** form:
   a. From the workspace navigation pane, select the **Configuration** workspace.
   b. Expand the **Incidents** folder.
   c. Expand the **Trap Server** folder.
   d. Select **Trap Forwarding Configuration**.

   e. Select the **Trap Forwarding Filters** tab.

   f. Do one of the following:

      - To create a new configuration, click the * New icon.

      - To edit an existing configuration, click the Open icon in the row representing the configuration you want to edit.

   g. On the form that opens, navigate to the **Filter Expressions** tab.

   h. Locate the **Filter Expressions** table.

   i. Do one of the following:

      - To add a Trap Forwarding Filter, click the * New icon.

      - To edit an existing Trap Forwarding Filter, click the Open icon in the row representing the configuration you want to edit.

2. Make your configuration choices (see table).

3. Click **Save and Close** to save your changes and return to the **Trap Forwarding Configuration** form.

SNMP Trap Forwarding Filter Expression Configuration

Attribute	Description
Trap Object Identifier (OID)	Enter the Trap Object Identifier (OID) pattern you want to use for the SNMP trap filter. Valid values include: • The entire SNMP trap OID value. For example: `.1.3.6.1.6.5.66.7.1225` • The SNMP trap OID value that includes a wildcard as a placeholder for the missing values. For example, to specify only the SNMP trap OID matching prefix: `.1.3.6.1.6.5.66.7.*`

Configure Trap Forwarding Destinations

Pre-requisite: Make sure you have used the NNMi nnmincidentcfg.ovpl command line utility to automatically create or update the incident configurations for the SNMP traps you want to include. See "Load SNMP Trap Incident Configurations" on page 771 for more information. You must also create the Trap Forwarding Filters for the SNMP traps you want to forward. See "Configure Trap Forwarding Filters" on page 1378 for more information.

The Trap Forwarding Destinations tab enables you to specify the servers to which you want to forward SNMP traps. For example, you can configure NNMi to forward traps to a remote server that receives SNMP traps, such as an HP Operations Manager server or another NNMi management server. See "Forward Traps to a Remote Server Example" on page 1385 for more information. Use this tab to also specify the Trap Forwarding Filters to be used for this destination.

(*NNMi Advanced*) If this NNMi management server is a Regional Manager in your environment, see also "Configure Forward to Global Manager Settings for an SNMP Trap Incident (NNMi Advanced)" on page 927.

Note: See "Manage Incoming SNMP Traps" on page 769 for information about the criteria NNMi uses to determine when to receive or discard traps.

To configure Trap Forwarding Destinations:

1. Navigate to the **Trap Forwarding Configuration** form.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Expand the **Incidents** folder.
   c. Expand the **Trap Server** folder.
   d. Select **Trap Forwarding Configuration.**
2. Select the **Trap Forwarding Destinations** tab.
3. Do one of the following:
   - To create an SNMP Trap Forwarding Destination configuration, click the ✱ New icon, and continue.
   - To edit an SNMP Trap Forwarding Destination configuration, click the Open icon in the row representing the configuration you want to edit, and continue.
   - To delete an SNMP Trap Forwarding Destination configuration, click the ✖ Delete icon.
4. In the "Trap Forwarding Destination Form" below, provide the required information.
5. Do one of the following:
   - To create an SNMP Trap Forwarding Filter configuration, click the ✱ New icon, and continue.
   - To edit an SNMP Trap Forwarding Filter configuration, click the Open icon in the row representing the configuration you want to edit, and continue.
   - To delete an SNMP Trap Forwarding Filter configuration, click the ✖ Delete icon.
6. In the "Destination Filter Form" on page 1384, provide the required information.
7. Click **Save and Close** to save your changes and return to the **Trap Forwarding Configuration** form.

The next time a trap that passes the Trap Forwarding Filter arrives, NNMi forwards the trap to the specified Trap Forwarding Destination.

Trap Forwarding Destination Form

Pre-requisite: Make sure you have used the NNMi nnmincidentcfg.ovpl command line utility to automatically create or update the incident configurations for the SNMP traps you want to include. See "Load SNMP Trap Incident Configurations" on page 771 for more information. You must also create the Trap Forwarding Filters for the SNMP traps you want NNMi to forward. See "Configure Trap Forwarding Filters" on page 1378 for more information.

The Trap Forwarding Destinations form enables you to specify the servers to which you want NNMi to forward SNMP traps.

Note: See "Manage Incoming SNMP Traps" on page 769 for information about the criteria NNMi uses to determine when to receive or discard traps.

To configure a Trap Forwarding Destination:

1. Navigate to the **Trap Forwarding Configuration** form:
   a. From the workspace navigation pane, select the **Configuration** workspace.
   b. Expand the **Incidents** folder.
   c. Select **Trap Forwarding Configuration**.
2. Select the **Trap Forwarding Destinations** tab.
3. Do one of the following:
   - To add an SNMP Trap Forwarding Destination configuration, click the New icon that precedes the configuration you want to edit, and continue.
   - To edit an SNMP Trap Forwarding Destination configuration, click the Open icon in the row representing the configuration you want to edit, and continue.
   - To delete an SNMP Trap Forwarding Destination configuration, click the Delete icon.
4. Make your configuration choices (see table).
5. Click **Save and Close** to save your changes and return to the **Trap Forwarding Configuration** form.

SNMP Trap Forwarding Destination Configuration

Attribute	Description
Destination Name	Enter the name you want to use for this SNMP Trap Forwarding Destination configuration. Alpha-numeric and special characters (~ ! @ # $ % ^ & * ( ) _+ -) are permitted. No spaces are permitted.
Destination Address	Enter the IP address for the destination server. (*NNMi Advanced*) You can use IPv4 or IPv6 addresses.
Destination Port	Enter the UDP port number for the destination server.
Forwarding Options	• **Default** - NNMi processes the trap before forwarding. Click here for more information. NNMi adds two new varbinds to SNMPv2 traps for storing origin address information: ▪ Origin IP Address ▪ Origin IP Address type **Tip:** NNMi does not add these varbinds to SNMPv1 traps because that information is in each SNMPv1 traps' PDU header IP Address field.

SNMP Trap Forwarding Destination Configuration, continued

Attribute	Description
	See "Trap Varbinds Provided by NNMi" on page 1386 for more information. • **SNMPv3 to SNMPv2c Conversion** - NNMi converts an incoming SNMPv3 trap to SNMPv2c. Click here for more information. When converting SNMPv3 traps to SNMPv2c traps, NNMi does the following: ▪ Includes a Context Name varbind - Contains the `contextName` from the original SNMPv3 trap. ▪ Creates a Community Name - The Context Engine ID and SNMPv3 User Name of the original SNMPv3 trap are combined as follows: `username@contextEngineID`. For example, `ciscoAdmin@8000000b7f3cbec5632b47455e97070c` See "Trap Varbinds Provided by NNMi" on page 1386 for more information. • **Original Trap (UNIX only/IPv4 only)** - NNMi forwards the trap without any changes under certain circumstances. Click here for more information. ▪ Only forwarded from NNMi management servers on UNIX operating systems. ▪ Only forwards traps received-from IPv4 sources and forwarded-to IPv4 destinations.
Specify the Trap Forwarding Filters to Use	Use the Trap Forwarding Filters form to specify the Trap Forwarding Filters configurations to use. These filters determine which traps NNMi forwards to the destination you specify.

Destination Filter Form

Pre-requisite: Make sure you have used the NNMi nnmincidentcfg.ovpl command line utility to automatically create or update the incident configurations for the SNMP traps you want to include. See "Load SNMP Trap Incident Configurations" on page 771 for more information. You must also create the Trap Forwarding Filters for the SNMP traps you want to forward. See "Configure Trap Forwarding Filters" on page 1378 for more information.

The Trap Forwarding Filter Form enables you to specify the Trap Forwarding Filters that you want to apply for the SNMP traps NNMi forwards to the specified Trap Forwarding Destination.

Note: See "Manage Incoming SNMP Traps" on page 769 for information about the criteria NNMi uses to determine when to receive or discard traps.

To configure the Trap Forwarding Filters:

1. Navigate to the **Trap Forwarding Filter** form:
   a. From the workspace navigation pane, select the **Configuration** workspace.
   b. Expand the **Incidents** folder.

c. Expand the **Trap Server** folder.

d. Select **Trap Forwarding Configuration**.

e. Select the **Trap Forwarding Destinations** tab.

f. Do one of the following:

   - To create a new configuration, click the New icon.
   - To edit an existing configuration, select a row, and click the Open icon in the row representing the configuration you want to edit.

g. On the form that opens, navigate to the **FIlter Expressions** tab.

h. Locate the **Filter Expressions** table.

i. To create a **Filter Expression**, click the New icon.

2. Make your configuration choices (see table).

3. Click **Save and Close** to save your changes and return to the **Trap Forwarding Configuration** form.

SNMP Trap Forwarding Filter

Attribute	Description
Filter	Click the Lookup icon. Select Open from the drop-down menu to view information about the selected Filter, if any. Select Quick Find to select the Trap Forwarding Filter you want to use for the current Trap Forwarding Destination.

Forward Traps to a Remote Server Example

Use this help topic to guide you when you want to forward traps to a remote server that can receive SNMP traps, such as an HP Operations Manager or another NNMi management server.

Note: See "Manage Incoming SNMP Traps" on page 769 for information about the criteria NNMi uses to determine when to receive or discard traps.

To configure NNMi to forward SNMP traps to a remote server:

1. Navigate to the **Trap Forwarding Configuration** form.

   a. From the workspace navigation panel, select the **Configuration** workspace.

   b. Expand the **Incidents** folder.

   c. Expand the **Trap Server** folder.

   d. Select **Trap Forwarding Configuration.**

2. Navigate to the **Trap Forwarding Filters** tab.

3. Click the **New** icon.

4. Enter a filter name; for example, `NNMi Remote Server`.
5. Navigate to the **Filter Expressions** tab.
6. Click the **New** icon.
7. In the Trap Object Identifier attribute, enter: `.1.3.*`
8. Click **Save and Close** to return to the **Trap Forwarding Filter** form.
9. Click **Save and Close** to return to the **Trap Forwarding Configuration** form.
10. Navigate to the **Trap Forwarding Destinations** tab.
11. Click the **New** icon.
12. Provide the following information for the remote server to which SNMP traps will be forwarded:
    a. Destination Name.
    b. Destination Address.
    c. Destination Port.
    d. Forwarding Options.
13. Click **Save and Close** to return to the **Trap Forwarding Configuration** form.
14. Click **Save and Close** to save your changes.

In this example, Network Node Manager i Software forwards any SNMP traps with the enterprise address `.1.3.*` to the trap destination you configured.

When forwarding SNMP traps, note the following:

- NNMi appends two varbinds to the original SNMP trap and forwards it to the configured destinations.
- Trap forwarding does not result in any NNMi Incident enrichment to the forwarded SNMP traps.

Trap Varbinds Provided by NNMi

NNMi provides the following varbinds for use when forwarding SNMP traps.

> **Note: Note**: NNMi does not create these varbinds if the Forwarding Options attribute is set to *Original Trap (UNIX only)* when configuring trap forwarding destinations. See "Trap Forwarding Destination Form" on page 1382 for more information.

SNMP Trap Varbinds Provided by NNMi

Name	oid	Type	Description
Origin IP address	.1.3.6.1.4.1.11.2.17.2.19.1.1.3	InetAddress	*SNMPv2 traps only*. Contains the IP address (v4 / v6) of the original SNMP notification that generated the trap.

SNMP Trap Varbinds Provided by NNMi, continued

Name	oid	Type	Description
			Tip: NNMi does not add this varbind to SNMPv1 traps because that information is in each SNMPv1 traps' PDU header IP Address field.
Origin IP Address type	.1.3.6.1.4.1.11.2.17.2.19.1.1.2	InetAddressType	*SNMPv2 traps only*. Contains the type of the IP address (v4 / v6) of the Original IP Address varbind. The value "1" indicates IPv4 and "2" indicates IPv6. **Tip:** NNMi does not add this varbind to SNMPv1 traps because that information is in each SNMPv1 traps' PDU header IP Address field.
Context Name	.1.3.6.1.4.1.11.2.17.2.19.1.1.1	SnmpAdminString	Contains the contextName present in the original SNMPv3 notification. This varbind is present only when NNMi converts an SNMPv3 notification to an SNMPv2c trap. See "Trap Forwarding Destination Form" on page 1382 and "Configure NNMi SNMPv3 Security Settings for Trap Forwarding and Inform-Requests" on page 1376 for more information.

Configure Trap Logging

NNMi enables you to configure the logging format for SNMP traps that you want to appear in the `trap.log` and `trap.csv` log files. You can also override these trap logging configurations on a Node Group basis. This feature is useful when you want to track your trap history as well as customize a trap's message format and resolve varbind values.

The `trap.log` and `trap.csv` files are located in the following directories:

Windows

%NnmDataDir%\log\nnm

UNIX

$NnmDataDir/log/nnm

See the "NNMi Incidents" chapter of the *HP Network Node Manager i Software Deployment Reference* for more information about configuring these file properties.

When configuring trap logging, you perform the following tasks:

- Use the Basics pane of the Configure Trap Logging form to configure Trap Logging.
- Optional. Configure Node Group Trap Configurations to override the Trap Logging Configuration on a Node Group basis.

Note: See "Manage Incoming SNMP Traps" on page 769 for information about the criteria NNMi uses to determine when to receive or discard traps.

Trap Logging Configuration Form

NNMi enables you to configure the logging format for SNMP traps that you want to appear in the `trap.log` and `trap.csv` log files. You can also override these trap logging configurations on a Node Group basis. This feature is useful when you want to track your trap history as well as customize a trap's message format and resolve varbind values.

The `trap.log` and `trap.csv` files are located in the following directories:

Windows

%NnmDataDir%\log\nnm

UNIX

$NnmDataDir//log\nnm

See the "NNMi Incidents" chapter of the *HP Network Node Manager i Software Deployment Reference* for more information about configuring these file properties.

Tip: To display the associated SNMP Trap Incident configuration, if any, use **Actions** > **Show SNMP Trap Configuration**.

To configure Trap Logging:

1. Navigate to the **Incidents** folder:
   a. From the workspace navigation pane, select the **Configuration** workspace.
   b. Expand the **Incidents** folder.
2. Expand the **Trap Server** folder.
3. Select **Trap Logging Configuration**.
4. Do one of the following:
   - To add a configuration, click the ✱ New icon, and continue.
   - To edit a configuration, double-click the row representing the configuration you want to edit, and continue.
   - To delete a configuration, select a row, and click the ✖ Delete icon.

5. Make your Basic configuration choices (see table).
6. Make your Log Configuration choices (see table).
7. Click **Save and Close** to save your changes and return to the previous form.

Note: See "Manage Incoming SNMP Traps" on page 769 for information about the criteria NNMi uses to determine when to receive or discard traps.

Trap Logging Basic Configuration

Name	Description
Name	The name is used to identify the logging configuration and must be unique. Use a name that will help you to remember the purpose or kind of SNMP trap for which you are configuring this log information. Valid characters include alphanumeric, dash (-), slash (/), colon (:), and underscore(_).
Trap Object ID	Specify the Object Identifier of the trap you want to log. You can obtain the OID value from the `trap.log` or `trap.csv` log file. **Note**: NNMi automatically logs the OID for any undefined traps to these files. Click here for more information about determining a trap OID for an undefined trap. 1. Export the `trap.csv` file to an Excel spreadsheet. 2. Search for `NO TRAP LOGGING CONFIGURATION FMT FOR`. **Tip**: This text should appear in the **Formatted Message** column. 3. Look for the trap OID value that follows this message. **Tip**: You can also navigate to the **OID** column to identify the OID for this trap.
Trap Logging	If the **Enabled** options is selected, NNMi logs this SNMP Trap to the `trap.log` and `trap.csv` log files for the nodes in the specified Node Group. If the **Disabled** option is selected, NNMi does not log the specified SNMP Trap configuration to the `trap.log` and `trap.csv` log files in the specified Node Group.

Trap Logging Log Configuration

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td>Log Message Format</td><td>Specify the information you want NNMi to include in the SNMP Trap's Message attribute value. You can use any combination of valid parameter strings and Custom Incident attributes to configure the Message.
<b>Note</b>: The Log Message limit is 1024 characters. If the returned values exceed this limit, NNMi truncates the value starting from the end of the returned text string.
For more information, see:
"Valid Parameters for Trap Logging Messages" on page 1401
"Include varbinds in Your Log Message Format " on page 1405</td></tr>
<tr><td>Use the SNMP Trap Incident Configuration values</td><td>Specifies that you want the values from the associated SNMP Trap Incident Configuration to be used for the following attributes:<ul><li>Severity</li><li>Category</li><li>Family</li><li>Incident Message Format</li></ul>If selected ☑, you are not able to provide values for the attributes listed</td></tr>
<tr><td>Severity</td><td>The Severity represents the seriousness calculated for the SNMP trap. Use the Severity attribute to specify the Severity that should be assigned to the SNMP trap when it appears in the <code>trap.log</code> and <code>trap.csv</code> log files.
Possible values are described in the following table.
<b>Incident Severity Values</b><table><tr><th>Attribute</th><th>Description</th></tr><tr><td><b>Normal</b></td><td>Indicates there are no known problems related to the associated object. This severity is meant to be informational. Generally, no action is needed for these incidents.</td></tr></table></td></tr>
</table>

Trap Logging Log Configuration, continued

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td></td><td>

Incident Severity Values, continued

<table>
<tr><th>Attribute</th><th>Description</th></tr>
<tr><td><b>Warning</b></td><td>Indicates there might be a problem related to the associated object.</td></tr>
<tr><td><b>Minor</b></td><td>Indicates NNMi has detected problems related to the associated object that require further investigation.</td></tr>
<tr><td><b>Major</b></td><td>Indicates NNMi has detected problems related to the associated object to be resolved before they become critical.</td></tr>
<tr><td><b>Critical</b></td><td>Indicates NNMi has detected problems related to the associated object that require immediate attention.</td></tr>
</table>

See Monitor Incidents for Problems for more information about these severity values.

</td></tr>
<tr><td>Category</td><td>

The Category attribute helps you organize your SNMP Traps. Select the category that you want to be associated with this SNMP Trap when it appears in the `trap.log` and `trap.csv` log files.

Each of the possible Category values is described in the following table.

Incident Categories Provided by NNMi

<table>
<tr><th>Category</th><th>Description</th></tr>
<tr><td><b>Accounting</b></td><td>Used to indicate problems related to usage statistics and allocation of costs associated with the billing of time and services provided by devices. This category is not used by NNMi with default configurations, but it is available for incidents you define.</td></tr>
<tr><td><b>Application Status</b></td><td>Indicates there is a problem with the health of the NNMi software. Examples of these kinds of events include license expiration (see "Extend a Licensed Capacity" on page 1575) or that a certain NNMi process or service lost</td></tr>
</table>

</td></tr>
</table>

Trap Logging Log Configuration, continued

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td></td><td>

Incident Categories Provided by NNMi, continued

<table>
<tr><th>Category</th><th>Description</th></tr>
<tr><td></td><td>connection to the Process Status Manager (see "Stop or Start an NNMi Process" on page 82 and "Stop or Start NNMi Services" on page 86).</td></tr>
<tr><td><b>Configuration</b></td><td>Indicates there is a problem with the configuration of a managed device. For example, there is a physical address mismatch.</td></tr>
<tr><td><b>Fault</b></td><td>Indicates a problem with the network, for example Node Down.</td></tr>
<tr><td><b>Performance</b></td><td>Indicates a Monitored Attribute value <i>crossed</i> a configured threshold. For example, Disk Space Utilization exceeds the configured threshold criteria for High Value = 90 percent .</td></tr>
<tr><td><b>Security</b></td><td>Indicates there is a problem related to authentication. For example, an SNMP authentication failure.</td></tr>
<tr><td><b>Status</b></td><td>Indicates some kind of status message. Examples of these kinds of incidents include "SNMP Link Up" or an "HSRP Group status Normal" message.</td></tr>
</table>

</td></tr>
<tr><td>Family</td><td>

You can use Family values to further categorize the types of SNMP Traps that might be generated. Select the Family that you want to be associated with this SNMP Trap when it appears in the `trap.log` and `trap.csv` log files.

Each of the possible Family values are described in the following table.

Incident Family Attribute Values Provided by NNMi

<table>
<tr><th>Family</th><th>Description</th></tr>
<tr><td><b>Address</b></td><td>Indicates the incident is related to an address problem.</td></tr>
<tr><td><b>Aggregated Port</b></td><td>Indicates the incident is related to a <b>Link Aggregation</b>[1] problem.</td></tr>
</table>

</td></tr>
</table>

[1] Protocols used on Switches to configure multiple Interfaces (Aggregation Member Interfaces) to function as if they were one (an Aggregator Interface). When two Aggregator Interfaces establish a connection, that connection is an Aggregator Layer 2 Connection. The Aggregator Layer 2 Connection appears on Layer 2 Neighbor View maps as a thick line with an Interface icon at each end (representing the Aggregator Interface). When you double-click the thick line, it converts into multiple thin lines representing the participating Aggregation Member Layer 2 Connections with their Aggregation Member Interfaces at each end of the lines.

Trap Logging Log Configuration, continued

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td></td><td>

Incident Family Attribute Values Provided by NNMi, continued

<table>
<tr><th>Family</th><th>Description</th></tr>
<tr><td><b>BGP</b></td><td>Indicates the incident is related to a problem with BGP (Border Gateway Protocol). This family is not used by NNMi with default configurations, but it is available for incidents you define.</td></tr>
<tr><td><b>Board</b></td><td>Indicates the incident is related to a board problem. This family is not used by NNMi with default configurations, but it is available for incidents you define.</td></tr>
<tr><td><b>Chassis</b></td><td>Indicates the incident is related to an board problem. This family is not used by NNMi with default configurations, but it is available for incidents you define.</td></tr>
<tr><td><b>Component Health</b></td><td>Indicates the incident is related to Node Component metrics collected by NNMi. See Node Form: Node Component Tab for more information about the Node Component metrics collected.</td></tr>
<tr><td><b>Connection</b></td><td>Indicates the incident is related to a problem with one or more connections.</td></tr>
<tr><td><b>Correlation</b></td><td>Indicates the incident has additional incidents correlated beneath it. These incidents are associated with a duplicate count so that you can determine the number of correlated incidents associated with it.</td></tr>
<tr><td><b>Custom Poller</b></td><td>Indicates the incident is related to the NNMi Custom Poller feature. See About Custom Poller.</td></tr>
<tr><td><b>HSRP</b></td><td><i>NNMi Advanced</i>. Indicates the incident is related to a problem with Hot Standby Router Protocol (<b>HSRP</b>[1]).</td></tr>
<tr><td><b>Interface</b></td><td>Indicates the incident is related to a problem with one or more interfaces.</td></tr>
<tr><td><b>License</b></td><td>Indicates the incident is related to a licensing problem.</td></tr>
<tr><td><b>NNMi Health</b></td><td>Indicates the incident is related to NNMi Health. See the Check NNMi Health for more information.</td></tr>
<tr><td><b>Node</b></td><td>Indicates the incident is related to a node problem.</td></tr>
<tr><td><b>OSPF</b></td><td>Indicates the incident is related to an OSPF problem. This family is not used by NNMi with default configurations, but</td></tr>
</table>

</td></tr>
</table>

[1] Hot Standby Router Protocol

Trap Logging Log Configuration, continued

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td></td><td>
<b>Incident Family Attribute Values Provided by NNMi, continued</b>
<table>
<tr><th>Family</th><th>Description</th></tr>
<tr><td></td><td>it is available for incidents you define.</td></tr>
<tr><td><b>RAMS</b></td><td><i>NNMi Advanced</i>. Indicates the incident is related to a Router Analytics Management System problem.</td></tr>
<tr><td><b>RMON</b></td><td>Indicates the incident is related to a Remote Monitor (IETF standard, RFC 1757) problem. This family is not used by NNMi with default configurations, but it is available for incidents you define.</td></tr>
<tr><td><b>RRP</b></td><td><i>NNMi Advanced</i>. Indicates the incident is related to a problem with a Router Redundancy Protocol configuration.</td></tr>
<tr><td><b>STP</b></td><td>Indicates the incident is related to Spanning-Tree Protocol problem. This family is not used by NNMi with default configurations, but it is available for incidents you define.</td></tr>
<tr><td><b>Syslog</b></td><td>NNMi does not use this Family with default configurations. It is available for incidents you define.</td></tr>
<tr><td><b>Trap Analysis</b></td><td>Indicates the incident is related to an SNMP trap storm.</td></tr>
<tr><td><b>VLAN</b></td><td>Indicates the incident is related to a problem with a virtual local area network.</td></tr>
<tr><td><b>VRRP</b></td><td><i>NNMi Advanced</i>. Indicates the incident is related to a problem with Virtual Router Redundancy Protocol (<b>VRRP</b>[1]).</td></tr>
</table>
</td></tr>
<tr><td>Incident Message Format</td><td>Displays the Message Format for the associated SNMP Trap Incident Configuration, if any.</td></tr>
<tr><td>Trap Enabled</td><td>Displays whether the associated SNMP Trap Incident Configuration, if any, is Enabled.</td></tr>
<tr><td>Author</td><td>See Author form for important information.
<b>Caution</b>: If the Author attribute value is <b>HP Network Node Manager</b>, any changes are at risk of being overwritten in the future.
Click the ▾ Lookup icon and select Show Analysis to display details about the currently selected Author, select Quick Find to access the list of existing Author values, or click ✱ New to create one.</td></tr>
</table>

[1] Virtual Router Redundancy Protocol

Node Group Logging Configuration Form

NNMi enables you override the Trap Logging Configuration for nodes in a specified Node Group.

To configure Node Group Trap Configuration:

1. Navigate to the **Incidents** folder:
   a. From the workspace navigation pane, select the **Configuration** workspace.
   b. Expand the **Incidents** folder.
2. Expand the **Trap Server** folder.
3. Select **Trap Logging Configuration**.
4. Navigate to the **Node Group Logging Configuration** tab.
5. Do one of the following:
   a. To add a configuration, click the New icon, and continue.
   b. To edit a configuration, double-click the row representing the configuration you want to edit, and continue.
   c. To delete a configuration, select a row, and click the Delete icon.
6. Make your Basic configuration choices (see table).
7. Make your Log Configuration choices (see table).
8. Click **Save and Close** to save your changes and return to the previous form.

Note: See "Manage Incoming SNMP Traps" on page 769 for information about the criteria NNMi uses to determine when to receive or discard traps.

Node Group Logging Basic Configuration

Name	Description
Node Group	Specifies the Node Group that contains the nodes for which you want to configure trap logging information. To specify a Node Group, click the Lookup icon, and do one of the following: • To display a list of possible Node Groups, select Quick Find. In the Quick Find dialog, select the Incident of interest. • To create a Node Group, select New.
Ordering	Ordering specifies the order in which the configuration should be considered for nodes that appear in multiple Node Groups and therefore might have conflicting Node Group Logging Configurations. NNMi uses the Node Group Logging Configuration that has the lowest Ordering value. For example, Ordering is used in the following scenario: • A node is in both the Routers Node Group and the Switches Node Group. • Node Group Logging Configuration is specified for both Node Groups.

Node Group Logging Basic Configuration, continued

Name	Description
	• The Ordering value for the Routers Node Group is 3. • The Ordering value for the Switches Node Group is 5 In this example, for any node that appears in both the Routers Node Group and the Switches Node Groups. NNMi uses the Node Group Logging Configuration specified for the Routers Node Group, which has the lowest Ordering number.
Logging	If the **Enabled** options is selected, NNMi logs this SNMP Trap to the `trap.log` and `trap.csv` log files for the nodes in the specified Node Group. If the **Disabled** option is selected, NNMi does not log the specified SNMP Trap configuration to the `trap.log` and `trap.csv` log files for nodes in the specified Node Group. If the **Inherited** option is selected, NNMi uses the **Logging** value from the **Logging Configuration** form. For example, if Logging is **Disabled** in the Logging Configuration for this trap, then logging is disabled for the nodes in the specified Node Group. This option is available only for **Node Group Logging Configuration**. See "Trap Logging Configuration Form" on page 1388for more information.

Trap Logging Log Configuration

Name	Description
Log Message Format	Specify the information you want NNMi to include in the SNMP Trap's Message attribute value. You can use any combination of valid parameter strings and Custom Incident attributes to configure the Message. For more information, see: "Valid Parameters for Trap Logging Messages" on page 1401 "Include varbinds in Your Log Message Format " on page 1405
Severity	The Severity represents the seriousness calculated for the SNMP trap. Use the Severity attribute to specify the Severity that should be assigned to the SNMP trap when it appears in the `trap.log` and `trap.csv` log files.

Trap Logging Log Configuration, continued

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td></td><td>

Note: If different from the associated SNMP Trap Incident Configuration, the Severity value overrides the SNMP Trap Incident Configuration Severity value.

Possible values are described in the following table.

Incident Severity Values

<table>
<tr><th>Attribute</th><th>Description</th></tr>
<tr><td><b>Normal</b></td><td>Indicates there are no known problems related to the associated object. This severity is meant to be informational. Generally, no action is needed for these incidents.</td></tr>
<tr><td><b>Warning</b></td><td>Indicates there might be a problem related to the associated object.</td></tr>
<tr><td><b>Minor</b></td><td>Indicates NNMi has detected problems related to the associated object that require further investigation.</td></tr>
<tr><td><b>Major</b></td><td>Indicates NNMi has detected problems related to the associated object to be resolved before they become critical.</td></tr>
<tr><td><b>Critical</b></td><td>Indicates NNMi has detected problems related to the associated object that require immediate attention.</td></tr>
</table>

See "Monitor Incidents for Problems" for more information about these severity values.

</td></tr>
<tr><td>Category</td><td>

The Category attribute helps you organize your SNMP Traps. Select the category that you want to be associated with this SNMP Trap when it appears in the `trap.log` and `trap.csv` log files.

Note: If different from the associated SNMP Trap Incident Configuration, the Category value overrides the SNMP Trap Incident Configuration Severity value.

Each of the possible Category values is described in the following table.

</td></tr>
</table>

Trap Logging Log Configuration, continued

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td></td><td>
<b>Incident Categories Provided by NNMi</b>
<table>
<tr><th>Category</th><th>Description</th></tr>
<tr><td><b>Accounting</b></td><td>Used to indicate problems related to usage statistics and allocation of costs associated with the billing of time and services provided by devices. This category is not used by NNMi with default configurations, but it is available for incidents you define.</td></tr>
<tr><td><b>Application Status</b></td><td>Indicates there is a problem with the health of the NNMi software. Examples of these kinds of events include license expiration (see "Extend a Licensed Capacity" on page 1575) or that a certain NNMi process or service lost connection to the Process Status Manager (see "Stop or Start an NNMi Process" on page 82 and "Stop or Start NNMi Services" on page 86).</td></tr>
<tr><td><b>Configuration</b></td><td>Indicates there is a problem with the configuration of a managed device. For example, there is a physical address mismatch.</td></tr>
<tr><td><b>Fault</b></td><td>Indicates a problem with the network, for example Node Down.</td></tr>
<tr><td><b>Performance</b></td><td>Indicates a Monitored Attribute value <i>crossed</i> a configured threshold. For example, Disk Space Utilization exceeds the configured threshold criteria for High Value = 90 percent .</td></tr>
<tr><td><b>Security</b></td><td>Indicates there is a problem related to authentication. For example, an SNMP authentication failure.</td></tr>
<tr><td><b>Status</b></td><td>Indicates some kind of status message. Examples of these kinds of incidents include "SNMP Link Up" or an "HSRP Group status Normal" message.</td></tr>
</table>
</td></tr>
<tr><td>Family</td><td>You can use Family values to further categorize the types of SNMP Traps that might be generated. Select the category that you want to be associated with this SNMP Trap when it appears in the <code>trap.log</code> and <code>trap.csv</code> log files.
<b>Note</b>: If different from the associated SNMP Trap Incident Configuration, the Family value overrides the SNMP Trap Incident Configuration Severity value.
Each of the possible Family values are described in the following table.</td></tr>
</table>

Trap Logging Log Configuration, continued

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td></td><td>
<b>Incident Family Attribute Values Provided by NNMi</b>
<table>
<tr><th>Family</th><th>Description</th></tr>
<tr><td><b>Address</b></td><td>Indicates the incident is related to an address problem.</td></tr>
<tr><td><b>Aggregated Port</b></td><td>Indicates the incident is related to a <b>Link Aggregation</b>[1] problem.</td></tr>
<tr><td><b>BGP</b></td><td>Indicates the incident is related to a problem with BGP (Border Gateway Protocol). This family is not used by NNMi with default configurations, but it is available for incidents you define.</td></tr>
<tr><td><b>Board</b></td><td>Indicates the incident is related to a board problem. This family is not used by NNMi with default configurations, but it is available for incidents you define.</td></tr>
<tr><td><b>Chassis</b></td><td>Indicates the incident is related to an board problem. This family is not used by NNMi with default configurations, but it is available for incidents you define.</td></tr>
<tr><td><b>Component Health</b></td><td>Indicates the incident is related to Node Component metrics collected by NNMi. See "Node Form: Node Component Tab" for more information about the Node Component metrics collected.</td></tr>
<tr><td><b>Connection</b></td><td>Indicates the incident is related to a problem with one or more connections.</td></tr>
<tr><td><b>Correlation</b></td><td>Indicates the incident has additional incidents correlated beneath it. These incidents are associated with a duplicate count so that you can determine the number of correlated incidents associated with it.</td></tr>
<tr><td><b>Custom Poller</b></td><td>Indicates the incident is related to the NNMi Custom Poller feature. See "About Custom Poller".</td></tr>
<tr><td><b>HSRP</b></td><td><i>NNMi Advanced</i>. Indicates the incident is related to a problem with Hot Standby Router Protocol (<b>HSRP</b>[2]).</td></tr>
<tr><td><b>Interface</b></td><td>Indicates the incident is related to a problem with one or more interfaces.</td></tr>
</table>
</td></tr>
</table>

[1] Protocols used on Switches to configure multiple Interfaces (Aggregation Member Interfaces) to function as if they were one (an Aggregator Interface). When two Aggregator Interfaces establish a connection, that connection is an Aggregator Layer 2 Connection. The Aggregator Layer 2 Connection appears on Layer 2 Neighbor View maps as a thick line with an Interface icon at each end (representing the Aggregator Interface). When you double-click the thick line, it converts into multiple thin lines representing the participating Aggregation Member Layer 2 Connections with their Aggregation Member Interfaces at each end of the lines.

[2] Hot Standby Router Protocol

Trap Logging Log Configuration, continued

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td></td><td>

Incident Family Attribute Values Provided by NNMi, continued

<table>
<tr><th>Family</th><th>Description</th></tr>
<tr><td><b>License</b></td><td>Indicates the incident is related to a licensing problem.</td></tr>
<tr><td><b>NNMi Health</b></td><td>Indicates the incident is related to NNMi Health. See the Check NNMi Health for more information.</td></tr>
<tr><td><b>Node</b></td><td>Indicates the incident is related to a node problem.</td></tr>
<tr><td><b>OSPF</b></td><td>Indicates the incident is related to an OSPF problem. This family is not used by NNMi with default configurations, but it is available for incidents you define.</td></tr>
<tr><td><b>RAMS</b></td><td><i>NNMi Advanced</i>. Indicates the incident is related to a Router Analytics Management System problem.</td></tr>
<tr><td><b>RMON</b></td><td>Indicates the incident is related to a Remote Monitor (IETF standard, RFC 1757) problem. This family is not used by NNMi with default configurations, but it is available for incidents you define.</td></tr>
<tr><td><b>RRP</b></td><td><i>NNMi Advanced</i>. Indicates the incident is related to a problem with a Router Redundancy Protocol configuration.</td></tr>
<tr><td><b>STP</b></td><td>Indicates the incident is related to Spanning-Tree Protocol problem. This family is not used by NNMi with default configurations, but it is available for incidents you define.</td></tr>
<tr><td><b>Syslog</b></td><td>NNMi does not use this Family with default configurations. It is available for incidents you define.</td></tr>
<tr><td><b>Trap Analysis</b></td><td>Indicates the incident is related to an SNMP trap storm.</td></tr>
<tr><td><b>VLAN</b></td><td>Indicates the incident is related to a problem with a virtual local area network.</td></tr>
<tr><td><b>VRRP</b></td><td><i>NNMi Advanced</i>. Indicates the incident is related to a problem with Virtual Router Redundancy Protocol (<b>VRRP</b>[1]).</td></tr>
</table>

</td></tr>
<tr><td>Author</td><td>

Caution: If the Author attribute value is **HP Network Node Manager**, any changes are at risk of being overwritten in the future.

- Click **Lookup** and select **Show Analysis** to display details about the currently selected Author.

</td></tr>
</table>

[1] Virtual Router Redundancy Protocol

Trap Logging Log Configuration, continued

Name	Description
	• Click **Quick Find** to access the list of existing Author values. • Click **New** to create an Author value.

Valid Parameters for Trap Logging Messages

When configuring Trap Logging messages, consider using SNMP Trap Incident information as part of the message. NNMi provides the following parameter values. Use these parameters as variables when formatting an incident message.

Tip: See the Using the Incident Form for more information about the parameter values.

Note: NNMi stores varbind values as custom incident attributes (CIAs).

See "Configure Trap Logging" on page 1387 for more information about configuring messages.

Parameter strings are available for the following:

Note: See the following tables to view the valid parameters for Trap Logging messages: Parameter Strings for all Incidents (Attributes from an Incident form), Parameter Strings for Node Source Objects (Attributes from a Node form), Parameter Strings for all Incidents (Attributes not Visible from any form), and Parameter Strings Compatible with NNMi 6.x/7.x.

- Parameter strings for all incidents (Incident form attributes) (Click here for a list of choices.)

Parameter Strings for all Incidents (Incident form attributes)

Parameter String	Description
$category, $cat	Value of the Category attribute in the Trap Logging Configuration. If no Trap Logging configuration is specified, NNMi uses the value `Configuration`.
$family, $fam	Value from the Family attribute in the Trap Logging Configuration. If no Trap Logging configuration is specified, NNMi uses the value `Node`
$lifecycleState, $lcs	By default, this value is `Registered.`
$name	Value of the Name attribute from the Trap Logging Configuration.
$nature, $nat	By default, this value is `Symptom`.
$origin, $ori	By default, this value is `SNMP Trap`.
$originOccurrenceTime, $oot	Value from the Origin Occurrence Time attribute in the associated SNMP Trap Incident Configuration. When trap arrived at the trap server
$priority, $pri	By default, this value is `None`.
$sev, $severity	Value of the Severity attribute of the Trap Logging Configuration. If no Trap Logging configuration is specified, NNMi uses the value `Normal`.

- Parameter Strings for Node Source Objects (Node form attributes) (Click here for a list of choices.)

Parameter Strings for Node Source Objects (Node form attributes)

Parameter String	Description
$managementAddress, $mga	The fully-qualified DNS name of the source address of the trap.
$sourceNodeLongName, $sln	The fully-qualified DNS name of the source address of the trap.
$sourceNodeName, $snn	The HostName of the trap source.

- Parameter Strings for all incidents (Additional information that is not visible in any form) (Click here for a list of choices.)

Parameter Strings for all Incidents (Attributes not visible in any form)

Parameter String	Description
$count, $cnt	Value representing the number of varbinds that appear in the SNMP Trap.
$oid	Value of the unique Object Identifier (OID) for the SNMP trap.
$originOccurrenceTimeMs, $oms	Time the trap was received in number of milliseconds (measured since January 1, 1970, 00:00:00 GMT - Greenwich Mean Time).

- Parameter Strings for all incidents (Additional information that is not visible in any form) (Click here for a list of choices.)

Parameter Strings Compatible with NNM 6.x/7.x

Parameter String	Description
$e	The value in Object Identifier (OID) format that specifies the Enterprise, if any
$ar, $aA, $aR	IP Address of the trap source node
$x	Date the trap was received . This date appears in Month Day, Year format; for example: October 28, 2011
$A	SNMP Agent address
$C	SNMP Read Community String for the source node
$E	The value in textual format that specifies the Enterprise, if any.
$F	Remote trap service (TrapServer) HostName.
$G	Generic trap type
$R	Hostname of the trap's source node
$S	Specific trap type
$T	sysUpTime
$V	Event type for the trap
$X	Time the trap was received in number of milliseconds (measured since January 1, 1970, 00:00:00 GMT - Greenwich Mean Time)

Information established in varbinds (Click here for a list of choices.)

Parameter Strings for varbinds

Parameter String	Description
$<position_number>	Value of the position number for a varbind . For example, to indicate you want to use the varbind in position 1, enter: `$1` If you know the varbind position number, use this parameter.
$<varbind_name>	Value of the name that is used for the varbind.
$<varbind_oid>	Value of the object identifier for a specified varbind. For example, `$.1.3.6.1.6.3.1.1.5.1.` Use this parameter when you are not certain of a varbind position number.
$*	Used to indicate you want all of the varbind values, to be passed to the action configuration.

- Functions to generate values (Click here for a list of choices.)

  The function described in the following table replaces the specified numeric value with the associated text value stored in the varbind.

  Note: The associated MIB must have been loaded using the nnmloadmib.ovpl command.

Functions to Generate Values Within the Incident Message

Function	Description
$oidtext ($<position_number>)	A <*position_number*> argument specifies the numeric value of the position number for a specific varbind . For example, `$oidtext($2).` **Note**: The position number you enter must represent a varbind that contains an Object Identifier (OID) value. NNMi returns the textual value of the OID for the varbind specified. Note the following: ▪ If the MIB is not loaded, NNMi returns the numeric OID value. ▪ If the OID has a MIB instance, the number representing the MIB instance is appended to the textual OID value.
$oidtext ($<varbind_oid>)	The <*varbind_oid*> argument specifies the Object Identifier (OID) for a specific varbind. For example, `$oidtext($.1.3.6.1.6.3.1.1.5.1.)` **Tip**: Use this argument to the $oidtext() function when you are not certain of a varbind position number. NNMi replaces the numeric value with the textual value of the OID you specify. Note the following:

Functions to Generate Values Within the Incident Message, continued

Function	Description
	▪ If the MIB is not loaded, NNMi returns the numeric OID value. ▪ If the OID has a MIB instance, the number representing the MIB instance is appended to the textual OID value.
$text ($<position_number>)	The <*position_number*> argument specifies the numeric value of the varbind position number. For example, to indicate you want to use the varbind in position 1, enter: `$1`. NNMi replaces the numeric value with the text value stored in the varbind. **Note**: If a text value is not available, NNMi returns the numeric value.
$text ($<varbind_oid>)	The <varbind_oid> argument specifies the object identifier for a specific varbind. For example, `$.1.3.6.1.6.3.1.1.5.1.` Use this argument to the $text function when you are not certain of a varbind position number. NNMi replaces the numeric value with the text value stored in the varbind. **Note**: If a text value is not available, NNMi returns the numeric value.

Include varbinds in Your Log Message Format

You can use varbinds in your message format to extend the amount of information presented. SNMP trap varbinds are identified by the Abstract Syntax Notation value (ASN.1).

To include a varbind in your Trap Logging Configuration message format, type the dollar-sign character (`$`) plus any of the following

- Varbind position number or asterisk (*) to include all varbind values
- Object identifier (oid) of the varbind (useful when the varbind position number is not consistent among vendors)

The following table presents some example formats with the subsequent output.

Example Incident Message Formats

Example Message Format	Output in Incident View
Possible trouble with $3	`Possible trouble with` <varbind 3>
Possible trouble with $11	`Possible trouble with` <varbind 11>
Possible trouble with $77 (where the varbind position 77 does not exist)	`Possible trouble with <Invalid or unknown varbind> 77`
Possible trouble with $3x	`Possible trouble with` <varbind 3>`x`
Possible trouble with $1.2.3.4.5	`Possible trouble with` <value of the varbind with oid of 1.2.3.4.5>

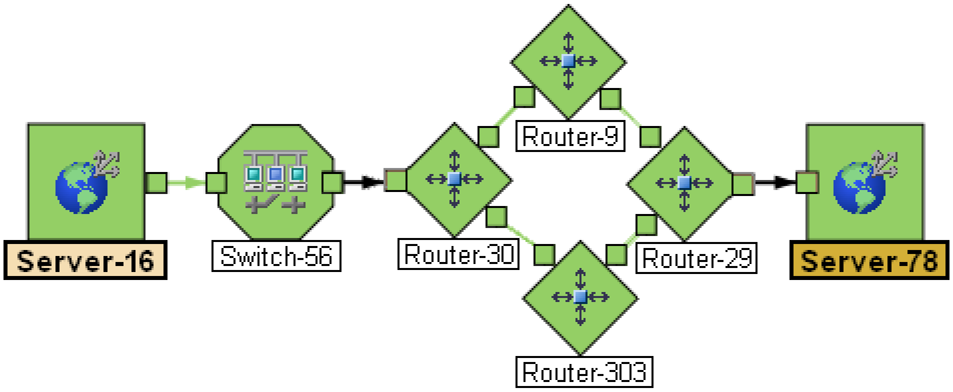
Tip: NNMi provides an error message when a varbind cannot be found. For example, if you enter an unavailable varbind position, name, or object identifier (oid), NNMi returns an "Invalid or unknown varbind" error message.

Chapter 16

Using Route Analytics Management Systems (RAMS) with NNMi Advanced

Route Analytics Management Systems (RAMS) is an IP Route Analytics tool that monitors routing protocols and builds a real-time routing topology map. You can use RAMS data to enhance NNMi.

- After configuring RAMS as described in "Configure HP Route Analytics Management Systems (NNMi Advanced)" on page 1409, the NNMi Path View displays the following enhanced information:
    - NNMi displays the Path View map faster, because RAMS does not use data collected from SNMP MIBs to determine the routing paths (avoiding any SNMP timeout issues).
    - Path View might be more accurate than the Path View data collected from NNMi alone.
    - When RAMS data determines the router paths, NNMi ignores the `PathConnections.xml file` (see "Configure a Path View Map" on page 498).
- After you configure RAMS as described in "HP RAMS MPLS WAN Configuration (NNMi Advanced)" on page 1410, NNMi provides the following additional information:
    - The Inventory workspace's MPLS WAN Clouds (RAMS) table view shows data. Additional information is provided on each MPLS WAN Cloud (RAMS) form.
    - A new NNMi map, the MPLS WAN Cloud Map view, is available from the Actions menu for participating objects (see MPLS WAN Cloud Map).
    - Path View shows all Equal Cost Multi-Paths (ECMP) rather than being limited to one route.



Note: *NNMi Advanced*. Path View works only with IPv4 addresses. The NNMi Advanced IPv6 address values are not valid choices for Path View. Any devices in your network that are configured with IPv6 addresses cannot be displayed on Path View maps.

For more information on MPLS WAN, see the *HP Route Analytics Management Software User's Guide*, which is available at: `[[Undefined variable nmVariables.selfsoveWebsite]]`.

Related Topics

Path Between Two Nodes

Path Calculation Rules

Path View Limitations

HP RAMS MPLS WAN *(NNMi Advanced)*

HP Route Analytics Management Software (RAMS) for MPLS WAN enables you to gather network connectivity information for enterprises that have multiple sites connected by a WAN through Internet Service Providers (ISPs). These ISPs use **MPLS**[1] within their own networks. MPLS enables the ISPs to support large numbers of Virtual Private Networks (VPNs). Although RAMS does not have visibility into the routing structure within the ISP network, it displays and analyzes routing topologies that extend across the WAN.

HP RAMS MPLS WAN is integrated with NNMi and is important if your enterprise has multiple sites that are connected by a Layer 3 VPN. Each of your sites will typically have one or more Customer Edge (CE) routers that are connected to the ISP's Provider Edge (PE) routers. The ISP handles all the routing (including **BGP**[2]), as well as the VPN tunneling through its own network. With MPLS WAN, you can use RAMS to monitor all the sites and provide enterprise connectivity information. The topology view shows how an enterprise site is connected to multiple sites through an MPLS WAN cloud.

Although detailed routing through the ISP is not available, RAMS indicates whether there is connectivity between the ISP's PE routers. When one of your sites advertises **routing prefixes**[3], you can determine whether the ISP is correctly passing all the routing prefixes (not dropping any or sending additional prefixes).

For more information on MPLS WAN, see the *HP Route Analytics Management Software User's Guide*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.

Related Topics:

"Configure HP Route Analytics Management Systems (NNMi Advanced)" on the next page

"Using Route Analytics Management Systems (RAMS) with NNMi Advanced" on the previous page

"HP RAMS MPLS WAN Configuration"

[1] Multiprotocol Label Switching
[2] Border Gateway Protocol
[3] A network protocol technique used to shorten or filter the amount of required routing information in each packet by declaring a prefix for an entire group of packets. This prefix also indicated the number of bits in the address.

Configure HP Route Analytics Management Systems (*NNMi Advanced*)

Route Analytics Management Systems (RAMS) is an IP Route Analytics tool that monitors routing protocols and builds a real-time routing topology map. RAMS data enhances the information available in NNMi Path View maps. See "Using Route Analytics Management Systems (RAMS) with NNMi Advanced" on page 1407for more information.

You can also use RAMS with the HP RAMS MPLS WAN feature, which enables you to gather network connectivity data between multiple sites connected by a WAN through Internet Service Providers (ISPs). See "Using Route Analytics Management Systems (RAMS) with NNMi Advanced" on page 1407for more information.

Note: *NNMi Advanced*. Path View works only with IPv4 addresses. The NNMi Advanced IPv6 address values are not valid choices for Path View. Any devices in your network that are configured with IPv6 addresses cannot be displayed on Path View maps.

To enable NNMi to use RAMS data, you must use the RAMS form to configure each RAMS server you want to use. The RAMS form provides details about the RAMS appliance and the associated RAMS database to be used with NNMi.

To configure a RAMS server:

1. Navigate to the **RAMS Servers** form.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Select **RAMS Servers**.
2. Do one of the following:
   - To establish a RAMS Server configuration, click the New icon and continue.
   - To edit a RAMS Server configuration, select a row, click the Open icon, and continue.
   - To delete a RAMS server configuration, select a row and click the Delete icon.
3. Provide the required information (see Basic Attributes table).
4. Click **Save and Close** to save your changes and return to the list of configured RAMS.

Basic Attributes

Attribute	Description
Host	Hostname (*not case-sensitive*) or IP address used to identify the RAMS appliance that you want NNMi to access.
Query Password	Query password configured for the RAMS appliance.
Database Name	Name of the database that NNMi should access. This database must reside on the RAMS appliance that you have identified in the Name attribute.
Priority	Used when you configure more than one RAMS appliance. Determines the order in which NNMi attempts to access the configured RAMS appliances. The lower the number, the higher the priority. For example, the number 1 is the highest priority.

Related Topics

"Using Route Analytics Management Systems (RAMS) with NNMi Advanced" on page 1407

"HP RAMS MPLS WAN Configuration (NNMi Advanced)" below

HP RAMS MPLS WAN Configuration *(NNMi Advanced)*

For more information on MPLS WAN, see the *HP Route Analytics Management Software User's Guide*, which is available at: `[[[Undefined variable nmVariables.selfsoveWebsite]]]`.

The HP NNMi – HP RAMS MPLS WAN integration provides actions for accessing several MPLS WAN tools from the NNMi console.

Enabling the HP NNMi – HP RAMS MPLS WAN Integration

This section describes the steps to enable the HP NNMi – HP RAMS MPLS WAN Integration.

Prerequisites:

- Configure the RAMS Server
- Create an NNMi Web Service Client for RAMS

To configure the connection between NNMi and the HP RAMS MPLS WAN, follow these steps:

1. In the NNMi console, open the HP NNMi – HP RAMS MPLS WAN Integration Configuration form:
   a. Select **Integration Module Configuration**
   b. Select **HP RAMS MPLS RAMS**
2. Select the **Enable Integration** check box to activate the integration fields on the form.
3. Enter the required information for connecting to the NNMi management server and to the RAMS server (see table)
4. Click **Submit**.
   The status message displays. If the status message indicates a problem connecting to the NNMi management server, click **Return**, and change the values as suggested in the message.

Changing the HP NNMi – HP RAMS MPLS WAN Integration Configuration

To change the connection between the NNMi and the HP RAMS MPLS WAN, follow these steps:

1. In the NNMi console, open the HP NNMi – HP RAMS MPLS WAN Integration Configuration form:
   a. Select **Integration Module Configuration**
   b. Select **HP RAMS MPLS RAMS**
2. Modify the configuration values as appropriate (see table)
3. Verify that the **Enable Integration** check box in the form is selected
4. Click **Submit**.
   The configuration settings are changed.

Disabling the HP NNMi – HP RAMS MPLS WAN Integration

To disable the connection between the NNMi and the HP RAMS MPLS WAN, follow these steps:

1. In the NNMi console, open the HP NNMi – HP RAMS MPLS WAN Integration Configuration form:
   a. Select **Integration Module Configuration**
   b. Select **HP RAMS MPLS RAMS**
2. Clear the **Enable Integration** check box
3. Click **Submit.**
   The integration fields are disabled and the changes take effect immediately.

HP NNMi – HP RAMS MPLS WAN Integration Configuration Form Reference

The HP NNMi – HP RAMS MPLS WAN Integration Configuration form contains the parameters for configuring communications between NNMi and RAMS. This form is available from the Integration Module Configuration workspace.

Note: Only NNMi users with the Administrator NNMi role can access the HP NNMi – HP RAMS MPLS WAN Integration Configuration form.

The following table lists the parameters for connecting RAMS to the NNMi management server:

Attribute	Description
NNMi Host	The fully qualified domain name of the NNMi management server. This field is pre-filled with host name that was used to access the NNMi console. Ensure that this value is the name that is returned by the `nnmofficialfqdn.ovpl -t` command run on the NNMi management server.
NNMi Port	The port for connecting to the NNMi console. This field is pre-filled with the NNMi port, as specified in the following file: **Windows**: `%NnmDataDir%\conf\nnm\props\nms-local.properties` **UNIX**: `/var/opt/OV/conf/nnm/props/nms-local.properties` For non-SSL connections, use the value of nmsas.server.port.web.http, which is 80 or 8004 by default (depending on the presence of another web server when NNMi was installed). For SSL connections, use the value of nmsas.server.port.web.http, which is 443 by default.
NNMi User	The NNMi User attribute value must be NNMi Web Services Client (used *only to provide access for software* that is integrated with NNMi). For information on Configuring the NNMi user interface, see User Groups Provided in NNMi.
NNMi Password	The password for the specified NNMi user.

, continued

Attribute	Description
RAMS MPLS WAN Rediscovery Interval (hours)	The time interval in hours to run the RAMS MPLS WAN discovery process.

Related Topics:

"Configure One or More Route Analytics Management Systems (NNMi Advanced)"

"Using Route Analytics Management Systems (RAMS) with NNMi Advanced"

HP RAMS and Global Network Management *(NNMi Advanced)*

HP Route Analytics Management Systems (RAMS) integrates with HP Network Node Manager i Software (NNMi) in a Global Network Management environment to enhance the Layer 3 network management.

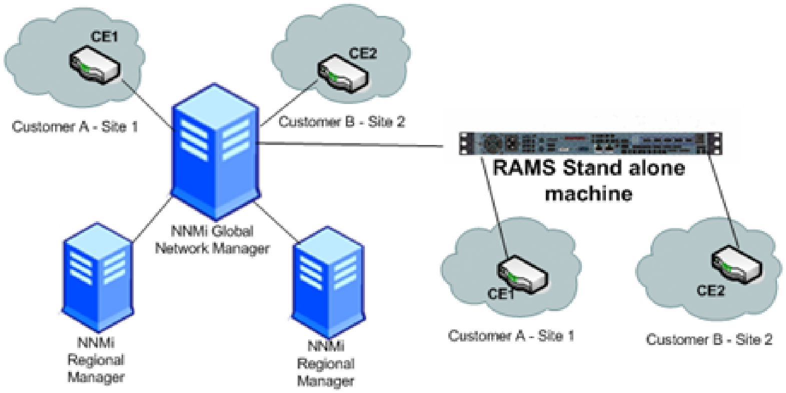
An HP RAMS device gathers the following information:

- Routes used for the data transmission
- Path computation
- Connectivity details of the geographically dispersed customer enterprises through a provider (MPLS WAN cloud)

NNMi integrates this information, resulting in a combined data view.

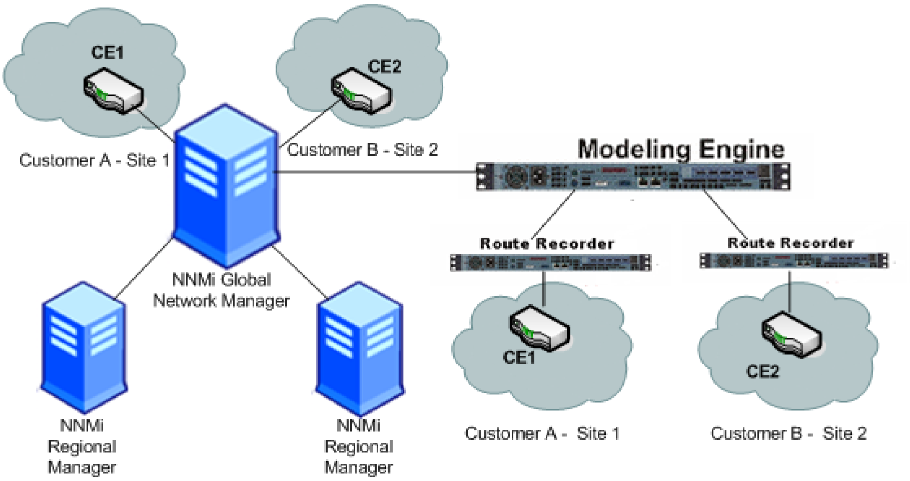
HP RAMS and NNMi integration can be setup in a Global Manger environment in one of the following three ways:

NNMi integrates with RAMS standalone



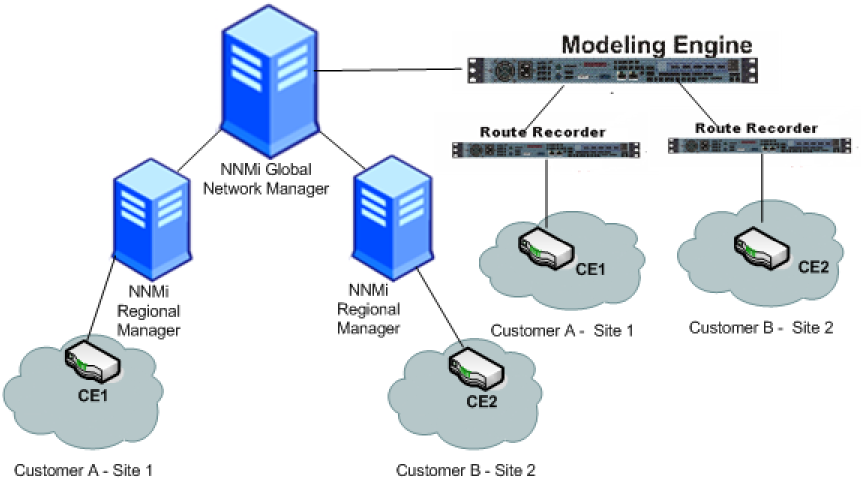
- NNMi Global Network Manager discovers CEs and displays Enhanced Virtual Private Network (EVPN) data
- NNMi receives MPLS WAN cloud information from RAMS standalone
- NNMi receives incidents from RAMS

NNMi integrates with the RAMS Modeling Engine (Distributed environment, with Customer Edges (CEs) discovered at Global Manager level)



- NNMi Global Network Manager discovers CEs and displays EVPN
- NNMi receives MPLS WAN cloud information from the RAMS Modeling Engine
- RAMS Modeling Engine receives information from different Route Recorders. Each Route Recorder discovers one CE to form the MPLS WAN cloud
- NNMi receives incidents from RAMS

NNMi integrates with the RAMS Modeling Engine (Distributed environment, with Customer Edges (CEs) discovered at Regional Manager level)



- NNMi Regional Manager discovers CEs. The complete EVPN displays at the NNMi Global Manager level
- NNMi Global Network Manager receives MPLS WAN cloud information from the RAMS Modeling Engine
- The RAMS Modeling Engine receives information from different Route Recorders. Each Route Recorder discovers one CE to form the MPLS WAN cloud
- NNMi receives incidents from RAMS

Chapter 17

Extending NNMi Capabilities

NNMi enables you to extend its capabilities in the following ways:

- You can integrate other programs into the console through the NNMi console menus. Plus much more, see "Control the NNMi Console Menus" below.
- You can work with MIB files to configure NNMi specifically for your environment. See "Managing MIBs" on page 1450.
- HP offers extended features, see "Purchase an HP Network Node Manager i Smart Plug-in" on page 1486.
- There are many ways to blend other software products into NNMi. See "Integrations with HP and Third-Party Products" on page 1487.
- "Add a Custom Attribute to a Node or Interface Object" on page 483

Tip: To extend NNMi's monitoring behavior, "Create Custom Polling Configurations" on page 419 so that NNMi monitors additional information using MIB expressions.

Control the NNMi Console Menus

NNMi enables you to configure the following menu items in the NNMi console menus:

- SNMP Line Graph Actions

  When you configure SNMP Line Graphs, you specify the graph appearance, including the MIB expression used to determine the values to be graphed. See "Configure SNMP Line Graph Actions" on page 1437 for more information.

- Launch Actions

  When you configure Launch Actions, you provide access to in-house tools, Web sites, or a variety of other resources. URLs are used to configure this powerful feature of NNMi. You must follow "W3C Rules for URLs" on page 1425. See "Configure Launch Actions" on page 1422 for more information. The syntax used to define the URL provides variables that can incorporate real-time data from the NNMi database. Click here for a list of choices:

- Java Actions provided by NNMi. See "Configure Java Actions" on page 1444
- JavaScript Actions provided by NNMi. See "Configure JavaScript Actions" on page 1443.

You control where each menu item appears in the menu structure:

- Choose an Ordering number for each menu item. See "Configure Menu Item Basic Details" on page 1417.
- Establish a nested structure of menu items. See "Create Menu Nesting" on the next page.

- Control when the menu item is available. See "Configure Menu Item Context Basic Details" on page 1420 and "Specify Optional Menu Item Enablement Filters" on page 1445.

Behavior of the Menu Items

If you do not assign an SNMP Line Graph or Launch Action to a particular menu item, the menu item never appears in an NNMi console menu.

Some Menu Item Actions require that a particular Object Type be selected for the menu item to be available. If the required Object Type is not selected, the color of the menu item turns from black to gray to indicate it is unavailable.

If you deselect the □ Enabled attribute on the Menu Item form, the menu item never appears in the NNMi console menu.

Create Menu Nesting

As an NNMi administrator, you configure how menu items are nested beneath the NNMi console menus. Menus can then contain menu items or other (cascading) menus.

To configure a Menu, beneath which other menu items can be nested:

1. Navigate to the **Menus** form.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Click to expand **User Interface**.
   c. Select **Menus**.
   d. Do one of the following:
      - To create a new menu, click the ✳ New icon, and continue.
      - To edit an existing menu, double-click the row representing the configuration you want to edit, and continue.
      - To delete a menu, select a row, and click the ✖ Delete icon.
2. Provide the required information to define the new Parent-level Menu Item (see basics table).
3. Click **Save and Close** to save and apply your changes.
4. Assign other menu items to appear beneath the Menu. The Menu label from step 2 is now available in the Parent Menu attribute drop-down list for Menu Items. See "Configure Menu Item Basic Details" on page 1417.
5. To test your changes to the menu:
   a. If required, access a view or form that contains the appropriate object type.
   b. If required, select an object instance.
   c. Select the menu you configured.
   d. Verify your changes are working.

Configuration Settings for a Menu Nesting

Attribute	Description
Menu Label	The text string that appears in the submenu. Ensure that your menu label is unique and provides an accurate indication about the group of submenu items that are found beneath this menu entry. If you add two Menu Labels with the same text string but different Unique Keys, both can show up beneath the menu you configured. The maximum length is 40 characters. Alpha-numeric, spaces, and underline characters are permitted.
Unique Key	**Caution**: This value cannot be changed after you click Save. Used as a unique identifier when exporting and importing menu definitions. To ensure that the value you enter is unique, it is recommended that you use the Java name space convention when providing this value. It is also useful to include the Menu Label value as part of the unique key as shown in the following example: `com.<company_name>.nnm.menu.<menu_label>` Type a maximum of 80 characters. Alpha-numeric and period characters are permitted. No spaces are permitted.
Author	Indicates who created or last modified the Menu nesting object. **Caution:** If the Author attribute value is **HP Network Node Manager**, any changes are at risk of being overwritten in the future. • Click **Lookup** and select **Show Analysis** to display details about the currently selected Author. • Click **Quick Find** to access the list of existing Author values. • Click **New** to create an Author value.
Parent Menu	Refine the nested location of this menu item. Click the Lookup icon next to the Parent Menu attribute, and do one of the following: • To select an existing parent-level menu item from the drop-down list (nesting the new menu item at a lower level in the menu structure), click the Quick Find icon. • To create a new parent-level menu item for nesting, click the New icon.
Ordering	A numeric value. NNMi checks for configuration settings in the order you define (lowest number first). NNMi uses the first match found to determine the placement of this menu item within the menu you configured. The Ordering numbers are calculated separately for each submenu (group of nested Actions menu items).

Configuration Settings for a Menu Nesting, continued

Attribute	Description
	Tip: It is recommended that ordering numbers are incremented by 10s to provide flexibility over time.
Prepend Separator	Used when a previous menu exists (based on the Ordering number). Inserts a separator line above the menu. Use this attribute to separate unrelated menus.
Enabled	Use to temporarily disable a Menu configuration (the Menu does not appear when disabled): **Enable** □ = Temporarily disable the selected configuration. **Enable** ☑ = Enable the selected configuration.

Configure Menu Item Basic Details

The **Menu Items** tab of the **User Interface Configuration** option enables you to make changes or additions to the items available in the NNMi console menus. For example, you can configure SNMP Line Graphs and provide menu items that display in-house tools, Web sites, or access a variety of other resources.

> **Note:** If you are configuring a Launch Action, you must select **Actions** for the Menu Label.

To make changes or additions to the items available in the NNMi console menus:

1. Navigate to the **Menu Items** view.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Click to expand **User Interface**.
   c. Select **Menu Items**.
   d. Do one of the following:
      - To create a new menu item, click the ✳ New icon, and continue.
      - To edit an existing menu item, double-click the row representing the configuration you want to edit, and continue.
      - To delete a menu item, select a row, and click the ✖ Delete icon.
2. Provide the required information to define the action (see Basics tables).

   > **Caution:** If you make changes to a Menu Item provided by NNMi, those changes are at risk of being overwritten in the future. See Author form for important information.

3. Provide the required Context details (see "Configure Menu Item Context Basic Details" on page 1420).
4. Click **Save and Close** to save and apply your changes.
5. To test your changes to the menu you configured:

a. If required, access a view or form that contains the appropriate object type.

b. If required, select an object instance.

c. Click the menu you configured.

d. Verify your changes are working.

Basics

Attribute	Description
Menu Item Label	The text string that appears as the menu link. Ensure that your menu label is unique and accurately reflects the intended use of the Menu Item. If you add two Menu Items with the same Menu Item Label string, both show up beneath the specified Parent Menu.
Unique Key	Used as a unique identifier when exporting and importing action definitions. To ensure that the value you enter is unique, it is recommended that you use the Java name space convention when providing this value. It is also useful to include the Menu Item Label value as part of the unique key as shown in the following example: `com.<company_name>.nnm.menu.item.<menu_item_label>` Type a maximum of 80 characters. Alpha-numeric and period characters are permitted. Spaces and underline characters are not permitted. **Caution:** This value cannot be changed after you click the Save icon.
Author	Indicates who created or last modified the Menu Item. **Caution:** If the Author attribute value is **HP Network Node Manager**, any changes are at risk of being overwritten in the future. • Click **Lookup** and select **Show Analysis** to display details about the currently selected Author. • Click **Quick Find** to access the list of existing Author values. • Click **New** to create an Author value.
Parent Menu	Specify where this action appears in the NNMi console: • Select any existing parent-level menu item from the drop-down list. • Create a new parent-level menu item. See "Create Menu Nesting" on page 1415 for more information.
Ordering	Valid entries are 1 to 100. This attribute controls where your menu item shows up in the list of available actions (lowest number appears at the top of the group of Menu Items).
Prepend Separator	Used when a previous menu item exists (based on the Ordering number). Inserts a separator line above the menu item. Use this attribute to separate unrelated menu

Basics, continued

Attribute	Description
	items.
Enabled	Use to temporarily disable a Menu Item configuration (when disabled, the Menu Item does not appear under the specified Parent Menu): **Enable** ☐ = Temporarily disable the selected configuration. **Enable** ☑ = Enable the selected configuration.

Selection

Attribute	Description
Selection Type	*Optional*. The default is Single Selection. The Menu Item is always available if you specify **No Selection** or **Any Selection**. • If you specify any of the following, an error message appears when the user launches the action before selecting an appropriate object or objects: ▪ **Any Selection** means zero or more selections required. ▪ **Single Selection** means exactly one selection required. ▪ **Multiple Selection** means one or more selections required. • If you specify **No Selection**, the user must launch the action without selecting any objects. An error message appears if any objects are selected.
Max Selection Count	*Only valid if Selection Type = Any Selection or Multiple Selection*. Zero means unlimited. Specify the maximum number of objects the user can select before launching this action.

Restrictions

Attribute	Description
Path View Only	If ☑ enabled, your action appears *only* in the Path View window's menu. See "Attributes per Object Type for Full URLs" on page 1426 for additional information about Path View Full URL configuration choices. If ☐ disabled, your action can appear in the menu of multiple views.

Description

Attribute	Description
Description	*Optional*. Provide a description of your action. Your description is visible only within this configuration form. Type a maximum of 1024 characters. Alpha-numeric, spaces, and special characters (~ ! @ # $ % ^ & * ( ) _+ -) are permitted.

Configure Menu Item Context Basic Details

NNMi enables you to configure NNMi console menu items using the Menu Item Context form.

To make changes or additions to the items available in the NNMi console menus:

1. Navigate to the **Menu Item Contexts** form.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Click to expand **User Interface**.
   c. Select **Menu Items** .
   d. Do one of the following:
      - To create a new menu item, click the ✳ New icon, and continue.
      - To edit an existing menu item, select a row, double-click the row representing the configuration you want to edit, and continue.
      - To delete a menu item, select a row, and click the ✖ Delete icon.
   e. Provide the Basics for this action (see "Configure Menu Item Basic Details" on page 1417).

      Caution: If the **Author** value is **HP Network Node Manager**, any changes are at risk of being overwritten in the future. See Author form for important information.
   f. Navigate to the **Menu Item Contexts** tab.
   g. Do one of the following:
      - To create a new Context configuration, click the ✳ New icon, and continue.
      - To edit an existing Menu Item Context configuration, double-click the row representing the configuration you want to edit, and continue.
      - To delete a Context configuration, select a row, and click the ✖ Delete icon.
2. Provide the Basic details for this Context configuration. (see the Basics table).
3. *Optional*. Limit the use of the menu item to a subset of the chosen object-type instances by defining filter criteria (see "Specify Optional Menu Item Enablement Filters" on page 1445
4. Click 💾 **Save and Close** to save and apply your changes.
5. To test your changes to the menu you configured:
   a. If required, access a view or form that contains the appropriate object type.
   b. If required, select an object instance.
   c. Click the menu you configured.
   d. Verify your changes are working.

Basics

Attribute	Description
Menu Item Action	Click the Lookup icon next to the Menu Item Action attribute and select one of the following: • Select Open to view the current configuration. There are four types of actions: ▪ "Configure Launch Actions" on the next page ▪ "Configure SNMP Line Graph Actions" on page 1437 ▪ "Configure JavaScript Actions" on page 1443 ▪ "Configure Java Actions" on page 1444 • Select ✱ **New SNMP Line Graph Action** to create a Line Graph. See "Configure SNMP Line Graph Actions" on page 1437 for more information. • Select ✱ **New Launch Action** to create an Launch Action menu item (access in-house tools, Web sites, or a variety of other resources). See "Configure Launch Actions" on the next page for more information.
Object Type	*Optional*. If you select **All Object Types**, your Graph Action or Launch Action is visible within the NNMi console menu in all views and forms. If you want your menu item to be available only within a view or form of a particular object type, select the desired Object Type from the drop-down menu. You can further limit the Action to a subset of object instances:
Required **NNMi Role**[1]	Specify the lowest NNMi Role allowed to access this action. From highest to lowest as follows: • Administrator • Operator Level 2 • Operator Level 1 • Guest • Web Service Client (*Only for software integrations* with NNMi. See "Integrations with HP and Third-Party Products" on page 1487) All User Groups associated with an NNMi Role that is a higher level than the NNMi Role you select can also access this action (see "Determine which NNMi User Group to Assign" on page 549). To determine the NNMi Role assigned to each User Group, in the **Configuration** workspace, expand the **Security** folder and select **User Groups**. For each User Group *provided by NNMi*, the **Description** attribute includes the NNMi Role associated with the User Group. (This setting cannot be modified in User Groups provided by NNMi.)

[1] Determined by your membership in one of four special NNMi User Groups (NNMi Administrators, NNMi Level 2 Operators, NNMi Level 1 Operators, or NNMi Guest Users). This membership determines what you can see and do within the NNMi console.

Basics , continued

Attribute	Description
	Caution: Each Tools and Action menu item provided by NNMi is associated with a *default NNMi Role*. (To determine the *default NNMi Role* assigned to each Action menu item, see "Actions Provided by NNMi" on page 43.) If you change the setting for a Menu Item provided by NNMi to a Role that is a *lower level Role* than the *default NNMi Role* assigned to the menu item, NNMi ignores that change. Any User Group with the lower level Role than the *default NNMi Role* cannot access the menu item.

Configure Launch Actions

The **Launch Actions** option enables you to configure Menu Items that will appear beneath the NNMi Actions menu. These additional menu items can access in-house tools, websites, or a variety of other resources.

To configure Launch Actions that users will access beneath the Actions menu:

> **Note:** You can configure a Launch Actions only for the **Actions** Parent Menu.

1. Navigate to the **Menu Item Contexts** form.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Click to expand **User Interface**.
   c. Select **Menu Items**.
   d. Do one of the following:
      - To edit an existing Launch Action menu item, double-click the row representing the configuration you want to edit, and continue.
      - To create a new Launch Action menu item, click the ✳ New icon, and continue.
      - To delete an Launch Action menu item, select a row, and click the ✖ Delete icon.

   e. Provide the Basic details for this Menu Item (see "Configure Menu Item Basic Details" on page 1417).

   > **Caution:** If the Author attribute value is **HP Network Node Manager**, any changes are at risk of being overwritten in the future. See Author form for important information.

   f. Navigate to the **Menu Item Contexts** tab.
   g. Do one of the following:
      - To edit an existing Menu Item Context, double-click the row representing the configuration you want to edit, and continue.

- To create a new Menu Item Context, click the ✳ New icon, and continue.
- To delete a Menu Item Context, select a row, and click the ✖ Delete icon.

2. Locate the **Menu Item Action** attribute. Click the Lookup icon, and click the ✳ **New Launch Action** icon.
3. In the **Full URL**attribute, type the URL and any required additional configuration syntax rules (see Launch Action Basics).
4. Click **Save and Close** to apply your changes and return to the Menu Item Context form.
5. Limit the use of the Action menu item to a subset of the chosen object-type instances by defining filter criteria (see "Specify Optional Menu Item Enablement Filters" on page 1445)
6. Click **Save and Close** to save and apply your changes.
7. To test your changes to the NNMi console menu:

   a. If required, access a view or form that contains the appropriate object type.

   b. If required, select an object instance.

   c. Click the menu you configured.

   d. Verify your changes are working.

   Troubleshooting Tip: If a specified attribute does not exist (for example, you made a mistake when typing the attribute's name), the attribute passes through literally (unresolved). For example:

   A node named "mynode" is selected, and the URL is:

   ```
   http://example.com?name=${name}&error=${error}
   ```

   The output would be:

   ```
   http://example.com?name=mynode&error=${error}
   ```

Launch Actions Basics

Attribute	Description
Name	Type a meaningful and descriptive name to help you remember the type of action.
Full URL	Add one or more definitions for the actual URL syntax. Type the full URL specification. The URL must comply with "W3C Rules for URLs" on page 1425. Include any required machine name and port number. Include any required parameters. • You can begin with either `http://` or `https://` For an example, click here: **To execute a Launch Action requesting something from NNMi**: `http://<serverName>:<portNumber>/nnm/launch?cmd=`**`showMenuItem&key=<MenuItemKey>[&nodename=<hostname or IP_address>`**

Launch Actions Basics, continued

Attribute	Description

	Note: If the NNMi Web server uses the https protocol, use `https` instead of `http`. See the "Working with Certificates for NNMi" chapter in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.
	`<serverName>` = the fully-qualified domain name of the NNMi management server (values allowed here are determined by the *Enable URL Redirect* setting in User Interface Configuration, see "Configuring the NNMi User Interface" on page 467)
	`<portNumber>` = the NNMi HTTP port number
	For a quick-reference list of all URL choices for launching NNMi, see **Help → Documentation Library → Integrate NNMi Elsewhere with URLs**. The Integrate NNMi Elsewhere with URLs page also contains a link to a list of sample URLs that you can copy/paste as a starting point.
	To execute a Launch Action requesting a script, application, or tool from your environment (not NNMi):
	`http://<serverName>:<portNumber>/<application>?<yourURLparameter1>=${<attribute>}&<yourURLparameter2>=${<attribute>}`
	Note: To extend the NNMi environment with additional applications, you must deploy them into a separate web-server or application-server on the same or different physical server from where the NNMi web-server or application-server is installed. See the *NNMi Developer's Toolkit* for more information.
	`<serverName>` = the appropriate fully-qualified domain name
	`<portNumber>` = the appropriate port number
	• You can also use other common URL protocols such as `ftp://`, `mailto://`, `news://`, or `telnet://`.
	• The list of available parameters changes depending on which limiting factors you configure. The **&** is used as the separator between the *`<yourURLparameter>`* and `${<attribute>}` pairs.
	For an example, click here:
	`http://example.com/nodeReport.jsp?myNode=${hostname}&mySnmpOid=${systemObjectId}`
	If the application that your URL calls is installed on the NNMi management server, the syntax can be as follows: `/<application>?<yourURLparameter1>=${<Attribute>}&<yourURLparameter2>=${<Attribute>}`

Launch Actions Basics, continued

Attribute	Description
	- "Attributes per Object Type for Full URLs" on the next page (Limits the availability of the Action to a subset of one object type.) - "Database Object Identifiers for Full URLs" on page 1435 (Limits the availability of the Action to *one specific instance* of an object.) - "Capability Attributes in Full URLs" on page 1430 (Limits the availability of the Action to a subset of objects.) - "Custom Attributes in Full URLs" on page 1431 (Limits the availability of the Action to a subset of objects.) - "Custom Incident Attributes (CIAs) in Full URLs" on page 1433 (Limits the availability of the Action to a subset of Incidents.) See "Attributes per Object Type for Full URLs" on the next page for more information about the valid attributes per Object Type.
Enable Cumulative Launch	If ☑ enabled, any object attribute references in the Full URL are populated with values from all selected objects. The multiple values are separated by a comma character. For example, if the attribute is "name", the URL results would be "name1,name2,name3". If □ disabled, the action launches a separate web page instance for each selected object. See "Attributes per Object Type for Full URLs" on the next page for details about including object attributes in your Full URL.
Browser Width	*Optional*. When empty, the default browser settings are used. If the value is 1 or more, the browser is launched this number of pixels wide.
Browser Height	*Optional*. When empty, the default browser settings are used. If the value is 1 or more, the browser is launched this number of pixels high.
Add Browser Decorations	If ☑ enabled, the web browser toolbar and menus appear when a user launches your URL. If □ disabled, the web browser has no toolbar or menu when a user launches your URL.

W3C Rules for URLs

The World Wide Web Consortium (W3C) allows only ASCII characters in URLs.

When configuring URLs, the following characters are always allowed:

- Alpha-numeric (A-Z a-z 0-9)
- - (hyphen)
- . (period)
- _ (underline)
- ~ (tilde)

Depending on the browser and the context, some characters require special formatting with Percent Encoding. A small number of possible values are shown in the quick reference table below.

You can designate the space character several ways:

- `+` (works in all browsers, recommended because it is easiest to read)
- `%20` (Percent Encoded value, works in all browsers)
- space character (works in the browsers supported by NNMi, but is not guaranteed to work in all browsers)

RFC 3986 Characters Reserved as Delimiters
(If not specifying a delimiter, use Percent-Encoding value)

Character	:	/	?	#	[	]	@	!	$
Percent Encoded	%3A	%2F	%3F	%23	%5B	%5D	%40	%21	%24
Character	&	'	(	)	*	+	,	;	=
Percent Encoded	%26	%27	%28	%29	%2A	%2B	%2C	%3B	%3D

Additional Commonly Used Characters and Their Percent Encoding

Character	space	%	<	>
Percent Encoded	%20 (or + allowed)	%25	%3C	%3E

Attributes per Object Type for Full URLs

There are a variety of methods to limit Launch Actions:

${<*attribute*>} values can be included in the Full URL syntax for each Object Type. For example:

- To limit the use of an Action available only from an Interface form, include ${ifAlias} as an ${<*attribute*>} value.
- To limit the use of an Action available only from a Node form, specify hostname: http://${hostname}:<*portNumber*>/<*application*>?attributeName1=${sysContact}&attributeName2=${sysName}

The following list includes the possible ${<*attributes*>} that can be included in the Full URL for each Object Type: For information about the complete Full URL syntax, see "Configure Launch Actions" on page 1422.

Interface [parameter list for interface]

${capabilities[capability.key=<*UniqueKey*>].capability.key} <value of one specific Capability, see "Capability Attributes in Full URLs" on page 1430 for more information>
${customAttributes[name=<*yourAttrName*>].value} <value of the matching Custom Attribute, see "Custom Attributes in Full URLs" on page 1431 for more information>
${ifAlias} <value from the ifAlias attribute>
${ifDescr} <value from the ifDescription attribute>
${ifIndex} <value from the ifIndex attribute>
${ifName} <value from the ifName attribute>
${ifType.label} <value from the ifType attribute>
${journal.notes} <value from the Notes attribute>
${managementMode} <value from the Management Mode attribute>
${name} <value from the Name attribute>
${overallStatus.lastChange} <value from the Status Last Modified attribute>
${overallStatus.status} <value from the Status attribute>
${physicalAddress} <value from the Physical Address attribute>
${speed} <value from the ifSpeed attribute>
Access any attribute on the related Node form, for example:
${hostedOn.hostname} <value from the Hosted On attribute, source Node's Hostname attribute>
${hostedOn.name} <value from the source Node's Name attribute>
Access an attribute on the related Node's Device Profile form:
${deviceProfile.devCategoryInterface}<value from the Category attribute's Label value>
${deviceProfile.devFamilyInterface}<value from the Family attribute's Label value>
${deviceProfile.devVendorInterface}<value from the Vendor attribute's Label>
Access an attribute on the related SNMP Agent form:
${hostedOn.snmpAgent.id} <value from the source Node's SNMP Agent Id attribute>${hostedOn.snmpAgent.agentSettings.agentEnabled} <value from the source Node's SNMP Agent Enabled attribute>

Interface Group [parameter list for interfaceGroup]

${name} <value from the Name attribute>
${notes} <value from the Notes attribute>
${nodeGroup.name} <value from the Node Group attribute>

Node [parameter list for node]

${capabilities[capability.key=<*UniqueKey*>].capability.key} <value of one specific Capability, see "Capability Attributes in Full URLs" on page 1430for more information>
${customAttributes[name=<*yourAttrName*>].value} <value of the matching Custom Attribute, see "Custom Attributes in Full URLs" on page 1431 for more information>
${hostname} <value from the Hostname attribute>
${journal.notes} <value from the Notes attribute>
${managementMode} <value from the Management Mode attribute>
${name} <value from the Name attribute>
${overallStatus.lastChange} <value from the Status Last Modified attribute>
${overallStatus.status} <value from the Status attribute>
${systemContact} <value from the System Contact attribute>
${systemDescription} <value from the System Description attribute>
${systemLocation} <value from the System Location attribute, the current value of the sysLocation

MIB variable>
${systemName} <value from the System Name attribute>
${systemObjectId} <value from the System Object ID attribute>
Access an attribute on the related Device Profile form:
${deviceProfile.deviceModel} <value from the Device Model attribute>
${deviceProfile.SNMPObjectID}<value from the SNMP Object ID attribute>
${deviceProfile.devCategoryNode}<value from the Category attribute's Label value>
${deviceProfile.devFamilyNode}<value from the Family attribute's Label value>
${deviceProfile.devVendorNode}<value from the Vendor attribute's Label value>
Access an attribute on the related SNMP Agent form:
${snmpAgent.id} <value from the Id attribute>
${snmpAgent.agentSettings.managementAddress} <value from the Management Address attribute>
${snmpAgent.agentSettings.agentEnabled} <value from the Agent Enabled attribute>
Access an attribute on the related Security Group form:
${securityGroup.name} <value from the Name attribute>
${securityGroup.uuid} <value from the UUID attribute>
Access an attribute on the related Tenant form:
${tenant.name} <value from the Name attribute>
${tenant.uuid} <value from the UUID attribute>

Node Group [parameter list for nodeGroup]

${name} <value from the Name attribute>
${notes} <value from the Notes attribute>
${overallStatus.lastChange} <value from the Status Last Modified attribute>
${overallStatus.status} <value from the Status attribute>

Incident [parameter list for incident]

Note: You cannot use the ${hostedOn.hostname} and ${customAttributes [name=<yourAttrName>].value} in the same Full URL.

${category.label} <value from the Category attribute>
${cias[name=<*cia.name*>].value} <value of one specific Custom Incident Attribute, see "Custom Incident Attributes (CIAs) in Full URLs" on page 1433 for more information>
${duplicateCount} <value from the Duplicate Count attribute>
${family.label} <value from the Family attribute>
${formattedMessage} <value from the Message attribute>
${getAttrOrName(<*attribute*>)} <value of the specified attribute of the Node associated with the Incident (if the Node exists in the database) or the *sourceNodeName* attribute of the Incident (if the Node was deleted from the database or never existed in the database). For example, ${getAttrOrName(hostname)}>
${journal.notes} <value from the Notes attribute>
${lifecycleState.label} <value from the Lifecycle State attribute>
${nature} <value from the Correlation Nature attribute>
${nodeUuid} <value of the uuid for the Source Node, see "Database Object Identifiers for Full URLs" on page 1435>
${nodeUuid.id} <value of the id for the Source Node, see "Database Object Identifiers for Full URLs" on page 1435>
${notes} <value from the Correlation Notes attribute>
${origin} <value from the Origin attribute>
${priority.label} <value from the Priority attribute>
${registration.created} <value from Created attribute>

${registration.modified} <value from the Last Modified attribute>
${severity} <value from the Severity attribute>
${sourceName} <value from Name attribute of the source object>
${sourceNodeName} <value from the Name attribute of the source object>
${sourceUuid} <value of the uuid for the Source Object, see "Database Object Identifiers for Full URLs" on page 1435>
${sourceUuid.id} <value of the source object's id attribute>
Access an attribute on the related source object form:
${sourceUuid.name} <value of the source object's Name attribute>
Access an attribute on the related Node form:
${nodeUuid.hostname} <<value from the source Node's Hostname attribute or IP address if no hostname is available>
${nodeUuid.name} <value of the Name attribute of the Source Node>

Layer 2 Connection [parameter list for layer2Connection]

${journal.notes} <value from the Notes attribute>
${name} <value from the Name attribute of the connection>
${source} <value of the Topology Source attribute, the protocol used to create the connection>

IP Address [parameter list for address]

${capabilities[capability.key=<*UniqueKey*>].capability.key} <value of one specific Capability, see "Capability Attributes in Full URLs" on the next page for more information>
${journal.notes} <value from the Notes attribute>
${managementMode} <value from the Direct Management Mode attribute>
${name} <value from the Name attribute>
${overallStatus.lastChange} <value from the Status Last Modified attribute>
${overallStatus.status} <value from the Status attribute>
${prefixLength} <value from the Prefix Length attribute>
${value} <value from the Address attribute>

IPSubnet [parameter list for subnet]

${journal.notes} <value from the Notes attribute>
${name} <value from the Name attribute>
${prefix} <value from the Prefix attribute>
${prefixLength} <value from the Prefix Length attribute>

Card [parameter list for card]

${capabilities[*CAPABILITY_NAME*]} <value of one specific Capability, see "Capability Attributes in Full URLs" on the next page for more information>
${capabilities[capability.key=<*UniqueKey*>].capability.key} <value of one specific Capability, see "Capability Attributes in Full URLs" on the next page for more information>
${entityPhysicalIndex} <value from the Physical Index attribute>
${firmwareVersion} <value from the Firmware Version attribute>
${hardwareVersion} <value from the Hardware Version attribute>
${hostingCard.name} <value from the Hosted On Card attribute>
${index} <value from the Index attribute>
${journal.notes} <value from the Notes attribute>
${managementMode} <value from the Management Mode attribute>
${modelName} <value from the Model Name attribute>
${monitoredAttributes.operationalState} <value from the Operational State attribute>
${overallStatus.status} <value from the Status attribute>

${overallStatus.lastChange} <value from the Status Last Modified attribute>
${redundantGroup.name} <value from the Redundant Group attribute>
${serialNumber} ${softwareVersion} <value from the Serial Number attribute>
${type} <value from the Type attribute>

Port [parameter list for port]

${associatedInterface.name} <value from the Associated Interface attribute>
${configuredDuplexSetting}<value from the Configured Duplex Setting attribute>
${index} <value from the Index attribute>
${journal.notes} <value from the Notes attribute>
${speed} <value from the Speed attribute>
${type} <value from the Type attribute>

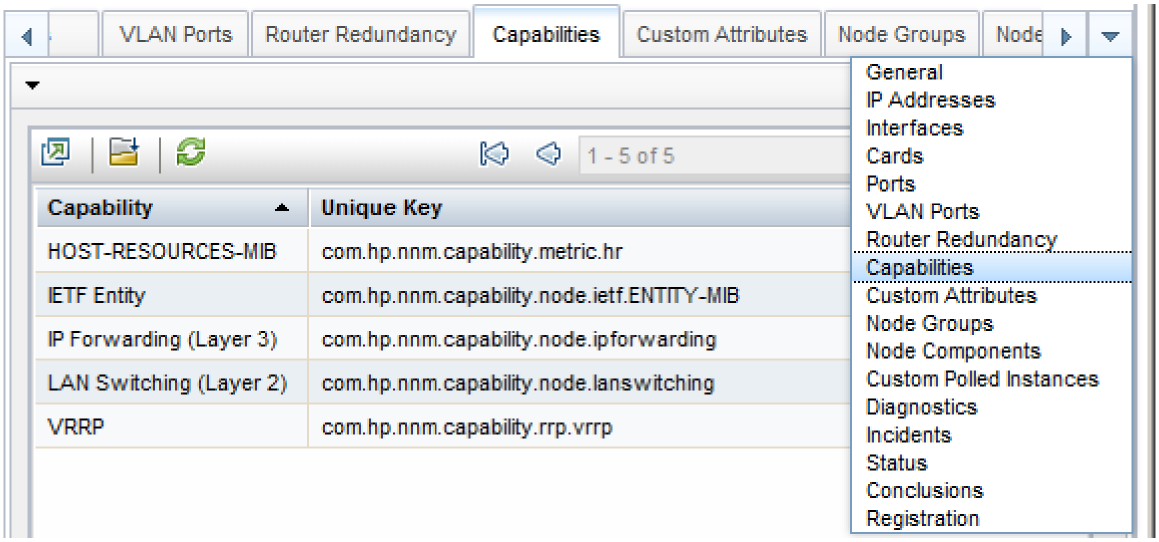
Capability Attributes in Full URLs

There are a variety of methods to limit Launch Actions:

NNMi node, interface, IP address, and card objects can have capability attributes:

Capabilities can be provided from HP Network Node Manager i Software Smart Plug-ins (iSPIs) or from integrations with other programs. See the documentation that came with any NNM iSPIs installed in your network environment.

To determine which group of capabilities are available for a specific object, navigate to a view for the object, select an instance of the object. Click the Open icon and navigate to the Capabilities tab. The items listed in the table are the Capabilities for that particular object instance. For example, the following illustration shows a Node form with three capability entries.



Capability	Unique Key
HOST-RESOURCES-MIB	com.hp.nnm.capability.metric.hr
IETF Entity	com.hp.nnm.capability.node.ietf.ENTITY-MIB
IP Forwarding (Layer 3)	com.hp.nnm.capability.node.ipforwarding
LAN Switching (Layer 2)	com.hp.nnm.capability.node.lanswitching
VRRP	com.hp.nnm.capability.rrp.vrrp

To pass Capability data within the Full URL, type (or copy and paste) the exact text string *from the object form, Capability tab*, **Unique Key** *attribute value*:

${**capabilities[capability.key=<*UniqueKeyValue*>].capability.key**}

Place the Capability into a location in the Full URL that enables the result your want:

> **Note:** If the NNMi Web server uses the https protocol, use `https` instead of `http`. See the "Working with Certificates for NNMi" chapter in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.

```
http://<serverName>:<portNumber>/<application>?<yourURLparameter1>=
${capabilities[capability.key= <UniqueKey_1>].capability.key}
&<yourURLparameter2>= ${capabilities[capability.key= <UniqueKey_
2>].capability.key}
```

> **Note:** To extend the NNMi environment with additional applications, you must deploy them into a separate web-server or application-server on the same or different physical server from where the NNMi web-server or application-server is installed. See the *NNMi Developer's Toolkit* for more information.

<serverName> = the appropriate fully-qualified domain name

<portNumber> = the appropriate port number

> **Note:** If the Capability that you request in the Full URL does not exist for the selected Node or Interface, the resulting URL passes an empty string.

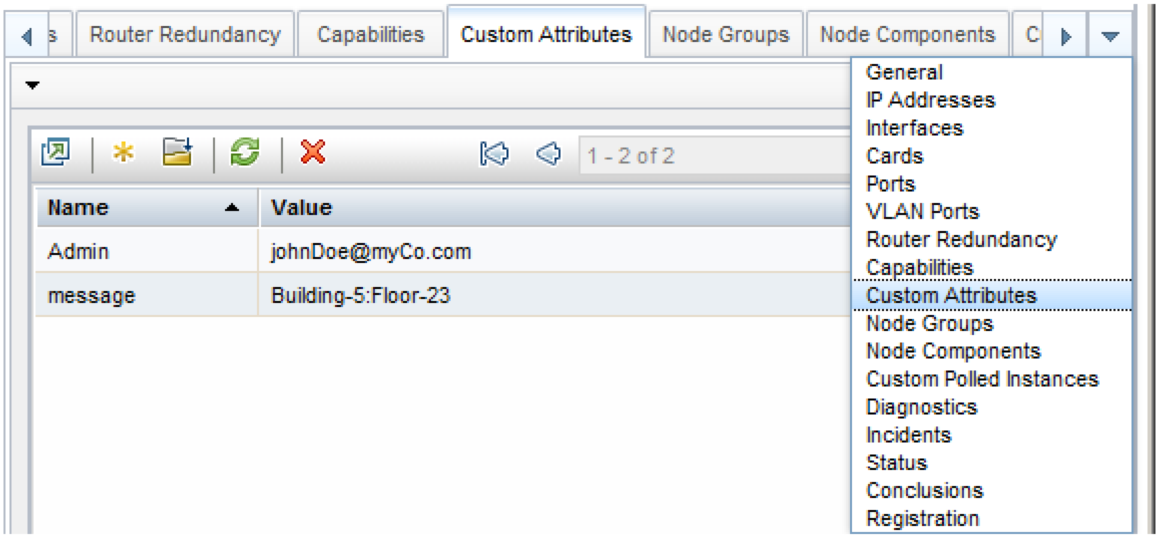
Custom Attributes in Full URLs

There are a variety of methods to limit Launch Actions:

Custom Attributes enable an NNMi administrator to add information to the Node object or Interface object. Custom Attributes can also be set by external applications that have been integrated with NNMi.

The Node form: Custom Attributes tab and Incident form: Custom Attributes tab display a table view of any Custom Attributes that have been added to the selected object. See "Add a Custom Attribute to a Node or Interface Object" on page 483.

To determine which group of Custom Attributes are available for a specific Node or Interface, navigate to a Node view or Interface view, select an instance of the object, click the Open icon and navigate to the Custom Attributes tab. The items listed in the table are the Custom Attributes for that particular node or interface. For example, the following illustration shows a Node form with two Custom Attribute entries.



Name	Value
Admin	johnDoe@myCo.com
message	Building-5:Floor-23

To pass Custom Attribute data within the Full URL, type (or copy and paste) the exact text string *from the Node or Interface form, Custom Attributes tab*:

`${`**`customAttributes[name=<yourAttrName>].value`**`}`

Place the Custom Attribute into a location in the Full URL that enables the result you want:

> **Note:** If you copy/paste this URL, remove the spaces that were added for line-ending purposes in the documentation.

```
http://<serverName>:<portNumber>/<application>?<yourURLparameter1>=
${customAttributes[value= <yourAttrValue>].name}&<yourURLparameter2>=
${customAttributes[name= <yourAttrName>].value}
```

> **Note:** To extend the NNMi environment with additional applications, you must deploy them into a separate web-server or application-server on the same or different physical server from where the NNMi web-server or application-server is installed. See the *NNMi Developer's Toolkit* for more information.

`<serverName>` = the appropriate fully-qualified domain name

`<portNumber>` = the appropriate port number

- Example 1:

  ```
  mailto:${customAttributes[name=Admin].value}?subject=URGENT Action
  Required&body=${customAttributes[name=message].value}&${hostname}
  router needs attention.
  ```

  Resulting URL:

  ```
  mailto:JohnDoe@myCompany.com?subject=URGENT Action
  Required&body=Building-5:Floor-23.&cisco4.myCo.com router needs
  attention.
  ```

- Example 2:

```
http://myCo.com/emailAdmin.jsp?name= ${hostname}&contact=
${customAttributes[name= Admin].value}&body= ${customAttributes
[name=message].value}
```

Resulting URL:

```
http://myCo.com/emailAdmin.jsp?name= cisco4.myCo.com&contact=
johnDoe@myCo.com&body= Building-5:Floor-23
```

> **Note:** If the Custom Attribute that you request in the Full URL does not exist for the selected Node or Interface, the resulting URL passes an empty string.

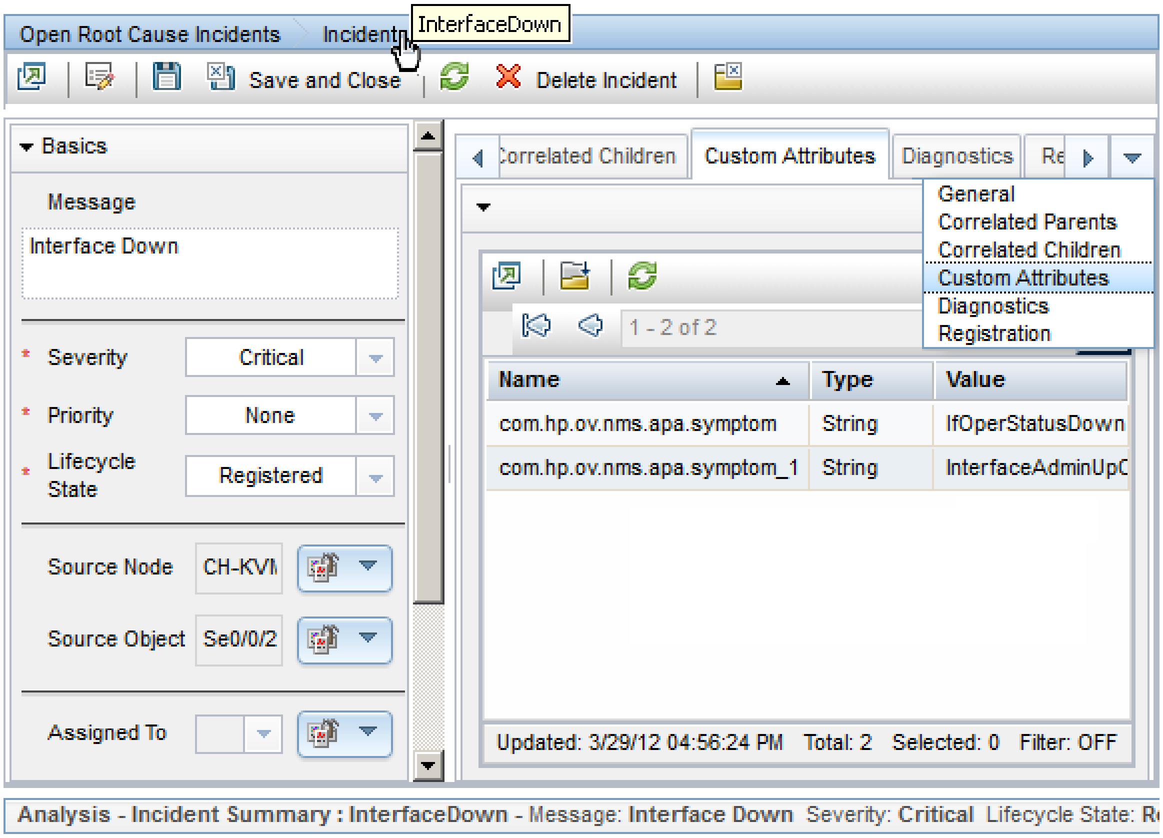
Custom Incident Attributes (CIAs) in Full URLs

There are a variety of methods to limit Launch Actions:

Custom Incident Attributes (CIAs) are used to provide the following types of information within incidents:

- SNMP trap varbinds identified by the Abstract Syntax Notation value, ASN.1 (Name = the MIB varbind identifier, Type = asn_*)
- Custom attributes provided by NNMi (Name = cia.*, Type=String). See "Custom Incident Attributes Provided by NNMi (Information for Administrators)" on page 647.

To determine which group of CIAs is available for a specific incident-type (for example, CiscoLinkDown), navigate to an Incident view, select an instance of that incident-type, click the Open icon and navigate to the Custom Attributes tab. The items listed in the table are the CIAs for that particular incident-type. For example, all CiscoLinkDown incidents would have the same group of CIAs shown in the illustration below.



Message: Interface Down

* Severity	Critical
* Priority	None
* Lifecycle State	Registered
Source Node	CH-KVI
Source Object	Se0/0/2
Assigned To	

Name	Type	Value
com.hp.ov.nms.apa.symptom	String	IfOperStatusDown
com.hp.ov.nms.apa.symptom_1	String	InterfaceAdminUpC

Updated: 3/29/12 04:56:24 PM Total: 2 Selected: 0 Filter: OFF

Analysis - Incident Summary : InterfaceDown - Message: Interface Down Severity: Critical Lifecycle State: R

To pass CIA data within the Full URL, type (or copy and paste) the exact text string *from the Incident form, Custom Attribute tab*, **Name** *attribute value*:

${**cias[name=<*cia_name*>].value**}

Place the CIA into a location in the Full URL that enables the result your want:

> **Note:** If the NNMi Web server uses the https protocol, use `https` instead of `http`. See the "Working with Certificates for NNMi" chapter in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.

```
http://<serverName>:<portNumber>/ <application>?<yourURLparameter1>=
${cias[name=<cia_name_1>].value}&<yourURLparameter2>= ${cias
[name=<cia_name_2>].value}
```

> **Note:** If you copy/paste this URL, remove the spaces that were added for line-ending purposes in the documentation.

> **Note:** To extend the NNMi environment with additional applications, you must deploy them into a separate web-server or application-server on the same or different physical server from where the NNMi web-server or application-server is installed. See the *NNMi Developer's Toolkit* for more information.

`<serverName>` = the appropriate fully-qualified domain name

`<portNumber>` = the appropriate port number

> **Note:** If the CIA that you request in the Full URL does not exist for the selected Incident, the resulting URL passes an empty string.

Database Object Identifiers for Full URLs

There are a variety of methods to limit Launch Actions:

If you need the Full URL to identify one specific record in the NNMi database, and find that it is not possible to provide a unique set of attribute values that distinguish that object instance from all other similar object instances, the *database unique identifiers* are valuable parameters.

The ID and UUID attributes are valid for all object types. NNMi displays the ID and UUID attribute values on the object form's Registration tab:

- `${uuid}` Universally Unique Object Identifier -Unique across all databases.
- `${id}` Unique Object Identifier - Unique across the Entire NNMi Database.

For example, the user can select an Interface object in the console, and use this Action to open the form of the Node in which the Interface resides:

```
/nnm/launch?cmd=showForm&objtype=Node&objid=${hostedOn.id}
```

Path View Attributes for Full URLs

There are a variety of methods to limit Launch Actions:

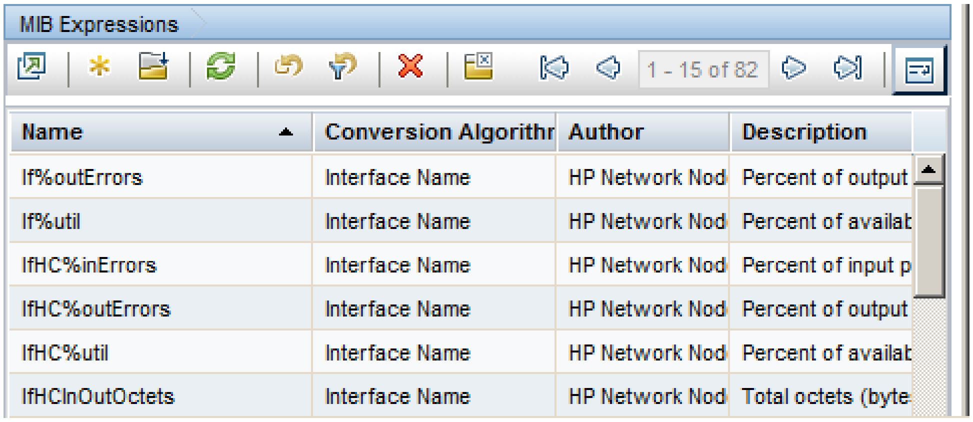
If you specified that a Launch Action appears only in the Path View menu, additional parameters are available:

${pathStartNodeName} <value of the Source attribute>
${pathEndNodeName} <value of the Destination attribute>
${pathList} <list of objects traversed along the path, separated by commas>
${pathCalculationDate} <date and time the path was calculated>

MIB Expressions in Full URLs

MIB Expressions enable an NNMi administrator to add SNMP MIB Expression information to a Graph.

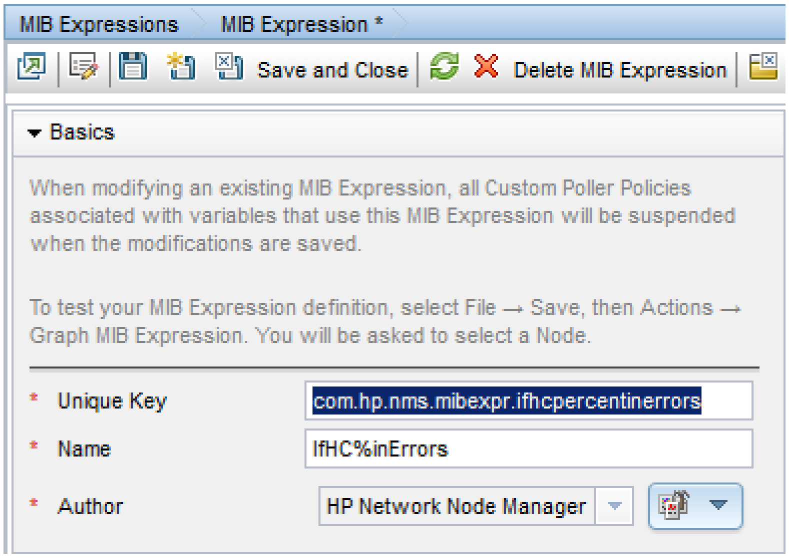
To determine the MIB Expressions available, navigate to the **MIB Expressions** option in the **Configuration** workspace. The items listed in the table are the MIB Expressions that have been created as shown in the following example:



Name	Conversion Algorithr	Author	Description
If%outErrors	Interface Name	HP Network Nod	Percent of output
If%util	Interface Name	HP Network Nod	Percent of availat
IfHC%inErrors	Interface Name	HP Network Nod	Percent of input p
IfHC%outErrors	Interface Name	HP Network Nod	Percent of output
IfHC%util	Interface Name	HP Network Nod	Percent of availat
IfHCInOutOctets	Interface Name	HP Network Nod	Total octets (byte

When using MIB Expressions in Graphs, provide the Unique Key value for the MIB Expression you want to use. To determine the Unique Key value, select the row containing the MIB Expression of interest, and click the Open icon. Look for the Unique Key value.

The following illustration shows the Basics section of a MIB Expression form with a Unique Key value provided by NNMi.



When modifying an existing MIB Expression, all Custom Poller Policies associated with variables that use this MIB Expression will be suspended when the modifications are saved.

To test your MIB Expression definition, select File → Save, then Actions → Graph MIB Expression. You will be asked to select a Node.

* Unique Key	com.hp.nms.mibexpr.ifhcpercentinerrors
* Name	IfHC%inErrors
* Author	HP Network Node Manager

To pass MIB Expression data within your Full URL, type (or copy and paste) the exact text string *from the Unique Key attribute* into the `expr=` parameter.

Place the `expr=[value]` into a location in your URL that enables the result you want as shown in the following example.

The following example displays a Line Graph of the percentage of input packets with errors for a selected interface.

> **Note:** The Unique Key value appears in bold. Replace space characters with "+" or %20 (see "W3C Rules for URLs" on page 1425).

```
http://<serverName>:<portNumber>/nnm/
launch?cmd=showLineGraph&init=ifindex=${ifIndex};
expr=com.mycompany.ifInerrors;/
```

```
label=Input+Errors;&title=Graph+SNMP+Interface+Input+Errors/
&objtype=SnmpAgent&objidlist=${hostedOn.snmpAgent.id}
```

> **Note:** If the NNMi Web server uses the https protocol, use `https` instead of `http`. See the "Working with Certificates for NNMi" chapter in the *HP Network Node Manager i Software Deployment Reference*, which is available at:
> `http://h20230.www2.hp.com/selfsolve/manuals`.

<serverName> = the fully-qualified domain name of the NNMi management server (values allowed here are determined by the *Enable URL Redirect* setting in User Interface Configuration, see "Configuring the NNMi User Interface" on page 467)

<portNumber> = the NNMi HTTP port number

> **Note:** If you copy/paste this URL, remove the spaces that were added for line-ending purposes in the documentation.

See "Attributes per Object Type for Full URLs" on page 1426 for more information.

Configure SNMP Line Graph Actions

The **User Interface Configuration** option enables you to configure Line Graphs that are available from the Actions menu. These graphs display real-time SNMP data for a selected node or interface. This feature is useful when you want to monitor a numeric MIB or MIB Expression value for a node or interface over a specified time interval. For example, you might want to monitor network traffic using the ifOutOctets MIB variable for a specified node. Or you might want to graph a MIB variable, such as Interface ifInOctets, to verify that a problem has been fixed for a specified interface before closing an incident.

Note: The node for which you want to display information must support SNMPv1, SNMPv2c, or SNMPv3.

NNMi provides a set of Line Graphs for nodes and for interfaces. See Line Graphs Provided by NNMi for more information.

To configure additional Line Graphs:

1. Navigate to the **Menu Items** form.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Click to expand **User Interface**.
   c. Select **Menu Items**.
   d. Do one of the following:
      - To create a Graph Action, click the ✳ New icon, and continue.
      - To edit an existing Graph Action, double-click the row representing the configuration you want to edit, and continue.
      - To delete a Graph Action, select a row, and click the ✖ Delete icon.

e. Provide the Basic details for this menu item (see "Configure Menu Item Basic Details" on page 1417).

   Caution: If the **Author** value is **HP Network Node Manager**, any changes are at risk of being overwritten in the future. See Author form for important information.

2. Select **Menu Item Contexts**.
3. Do one of the following:

   a. To create a Menu Item Context, click the ✱ New icon, and continue.

   b. To edit an existing Menu Item Context, double-click the row representing the configuration you want to edit, and continue.

   c. To delete a Menu Item Context, select a row, and click the ✖ Delete icon.
4. Provide the graph details for this Graph (see Basics table).
5. Provide the MIB Specification information (see "MIB Specification Form" on page 1440).
6. Click 🖫 **Save and Close** to save and apply your changes and return to the Menu Item Context form.
7. Limit the use of the Action menu item:
   - By object type (see "Configure Menu Item Basic Details" on page 1417).
   - By NNMi user role (see "Configure Menu Item Basic Details" on page 1417).
   - By defining a filter for a subset of the chosen object-type instances (see "Specify Optional Menu Item Enablement Filters" on page 1445).
8. Click 🖫 **Save and Close** to save and apply your changes.

   To test your changes to the Actions menu:

   a. If required, access a view or form that contains the appropriate object type.

   b. If required, select an object instance.

   c. Click the **Actions** menu.

   d. Verify your Graph is working.

Basics

Attribute	Description
Graph Title	Type a meaningful and descriptive title to display above the graph. The maximum length is 255 characters. Alpha-numeric characters, spaces, and periods are permitted.
Y-axis Label	Enter the text string to describe the Y-axis data displayed. NNMi displays this label vertically along the left-side of the Y axis. The maximum length is 255 characters. Alpha-numeric characters, spaces, and periods are permitted.

Basics, continued

Attribute	Description
	If you do not want to display a label for the Y-axis, leave this attribute blank.
Number of Lines	Specify the number of lines that will be initially displayed on the graph. An operator can display additional lines when viewing the Line Graph. To use the Default value specified in the User Interface Configuration, leave this attribute value blank. The default value that NNMi provides is 20.
Maximum Time Range (Hours)	The maximum time period in hours in which to retain the Line Graph data point sets. When the Maximum Time Range number is reached, NNMi discards the oldest data point sets so that it can display the most recent data for the time range you specify. For example, if you enter 24 hours, when 24 hours has passed, NNMi removes data starting with the initial data point set so that it can display data for the most recent 24-hour interval. Enter a decimal number indicating the maximum number of hours in which to retain the Line Graph data. If you specify 0 (zero), NNMi determines the best setting for the Maximum Time Range based on the Poling Interval specified.
Update Interval (Seconds)	The Update Interval in seconds to be used for collecting data to be displayed on the graph. **Note**: You can change the Update Interval for the current session when NNMi displays the graph. To use the Default value specified in the User Interface Configuration, leave this attribute value blank.
Fast Start	Select Fast Start when you want to increase the initial Polling Interval so that the initial data appears more quickly on the graph. When you select this option, NNMi increases the initial Polling Interval and then gradually decreases the Polling Interval until it reaches the Polling Interval configured for the graph.
Enable Cumulative Launch	If ☑ enabled, any object attribute references in the Full URL are populated with values from all selected objects. The multiple values are separated by a comma character. For example, if the attribute is "name", the URL results would be "name1,name2,name3". If ☐ disabled, the action launches a separate web page instance for each selected object. See "Attributes per Object Type for Full URLs" on page 1426 for details about including object attributes in your Full URL.
Browser Width	*Optional*. When empty, the default browser settings are used. If the value is 1 or more, the browser is launched with this number of pixels wide.
Browser	*Optional*. When empty, the default browser settings are used. If the value is 1

Basics, continued

Attribute	Description
Height	or more, the browser is launched with this number of pixels high.
Add Browser Decorations	If ☑ enabled, the web browser toolbar and menus appear when a user launches your URL. If □ disabled, the web browser has no toolbar or menu when a user launches your URL.

MIB Specification Form

The MIB Specification form enables you to indicate the following:

- The label to be displayed for each line that appears in the Line Graph Legend.
- The MIB Expression NNMi uses to gather the data shown in the graph.

To specify the Line Label and MIB Expression for an SNMP Line Graph Action:

1. Navigate to the **MIB Specification** form.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Click to expand **User Interface**.
   c. Select **Menu Items**.
   d. Do one of the following:
      - To create a new menu, click the ✱ New icon.
      - To edit a menu, double-click the row representing the configuration you want to edit.
   e. Navigate to the **Menu Item Contexts** tab.
   f. Do one of the following:
      - To create a new Context configuration, click the ✱ New icon.
      - To edit an existing Context configuration, double-click the row representing the configuration you want to edit.
   g. In the **Menu Item Context** form, locate the **Menu Item Action** attribute.
   h. Click the ▾ Lookup icon next to the **Menu Item Action** attribute, and do one of the following:
      - To create a new Line Graph, click the ✱ **New SNMP Line Graph Action** icon.
      - To edit the Line Graph associated with the Graph Action name displayed, double-click the row representing the configuration you want to edit.
   i. Provide the Basic details for this Graph Action (see the "Configure SNMP Line Graph Actions" on page 1437).
   j. Navigate to the **MIB Specifications** tab.

k. Do one of the following:

   - To create a new MIB Specification configuration, click the ✳ New icon.
   - To edit an existing MIB Specification configuration, double-click the row representing the configuration you want to edit.

2. Provide the Basic details for this MIB Specification configuration. (see the MIB Specification Basics table).

3. Click **Save and Close** to save and apply your changes.

MIB Specification Basics

Attribute	Description
Line Label	Enter the label that you want to be displayed for each line that appears in the Graph legend. Type a maximum of 40 characters. Alpha-numeric and special characters (~ ! @ # $ % ^ & * ( ) _+ -) are permitted. **Note**: When graphing multiple instances, the <*instance_string*> is appended to this value. See "MIB Expression Form (Line Graph)" on page 1474 for more information.
MIB Expression	Use this attribute to specify the MIB information that you want NNMi to poll. A MIB expression must include at least one MIB Variable. It can also include one or more of the following: • Constant • Arithmetic operator (+, -, *, /) If the MIB Expression does not include any arithmetic operators, valid types for any MIB Variable in the MIB Expression include the following: • Integer • Unsigned Integer • Octet String • Counter • Counter64 • Gauge • Time_Ticks If the MIB Expression contains any constants or arithmetic operators, the MIB Expression must evaluate to a numeric type. **Note**: If the MIB Expression is collecting a single MIB variable of type Time_Ticks, NNMi evaluates the return value as an Integer. Otherwise, it is treated as type Counter. When evaluating MIB expressions that include MIB variables of type Counter,

MIB Specification Basics , continued

Attribute	Description
	Counter64, or Time_Ticks, NNMi evaluates the MIB Variable using the difference in value between the most recent poll and the poll before it. If you want NNMi to calculate a rate over time in seconds, divide the MIB Expression by sysUptime. For example: `(((ifInOctets+ifOutOctets)*8/ifSpeed)*100)/sysUpTime*0.01` **Tip**: The sysUpTime variable is a value of hundredths of a second. When you want the rate in seconds, use `sysUpTime*0.01` in the MIB expression as shown in the previous example. **Note**: If you use a MIB variable of type Counter, Counter64, or Time_Ticks in the MIB Expression, NNMi automatically collects sysUpTime values if sysUpTime is not already in the MIB Expression. NNMi uses the sysUptime value to detect a system reboot. Any time a system reboot is detected, NNMi cannot determine the difference in values between polls for any Counter MIB variable and therefore does not calculate the MIB Expression for that poll. • If you select a MIB Variable from an Interface Table to include in the MIB Expression, note the following: ▪ *Line Graph Only*. When evaluating MIB Expressions that include MIB variables of type Counter or Counter64, NNMi requests both the high capacity and low capacity counter variable for any interface instance. If the high capacity Counter64 is enabled for any given interface instance, NNMi uses the high capacity counter. ▪ *Custom Poller Only*. When evaluating MIB Expressions that include MIB variables of type Counter, NNMi requests only the low capacity counter information for any interface instance. You create a MIB Expression by using the MIB Expression form. To access the MIB Expression form, click the Lookup icon and do one of the following: • Select **Quick Find** to select an existing MIB expression. • Select **Open** to edit the current MIB expression. • Select **New** to create a MIB expression. See "MIB Expression Form (Line Graph)" on page 1474 for information about using the MIB Expression form.
Instance Selection Algorithm	Used to specify how you want NNMi to handle instance discovery for Line Graphs that display multiple instances. Possible values are: • **All** - Use when you want NNMi to graph each instance of the object selected by the user. Note the following: ▪ When a node is selected, NNMi discovers all instances for that node, including the interfaces. When an interface is selected, NNMi graphs all selected interfaces.

MIB Specification Basics , continued

Attribute	Description
	▪ NNMi ignores any values entered in the Instance List attribute. ▪ When the Line Graph menu item is launched, NNMI populates `${snmpAgent.id}` and ${hostedOn.snmpAgent.id} with the ID values from the selected objects. The multiple values are separated by a comma character. ▪ NNMi displays a maximum of 100 instances. NNMi determines which 100 instances to display using the following calculation: 100 instances/(number of nodes selected)*(number of MIB expressions for the Action) • **Instance List** - Use when you want to specify the instances to be included in the Line Graph **Note**: You must specify the Instance List when using this option.
Instance List (Comma Separated)	Used to identify the instances to be graphed for an object. If your Menu Item Context is Node and you want to specify which nodes should be included on this Line Graph, enter the instance number for each of the node instances to be included on this Line Graph. For example, to graph CPU values, enter the instance number representing each CPU on the node, separated by commas. If your Menu Item Context is an Interface, these values are not used and the selected Interface(s)' `ifIndex` value is used as the SNMP instance.

Configure JavaScript Actions

NNMi provides a set menu items implemented with JavaScript. These menu items do not generate a new browser window.

Do not make any changes to these other than:

- Hide the menu item from the NNMi console (see the □ Enabled attribute "Configure Menu Item Basic Details" on page 1417).
- Change the Required Role setting (see "Configure Menu Item Context Basic Details" on page 1420).

Caution: If you make changes to a Menu Item provided by NNMi, those changes are at risk of being overwritten in the future. See Author form for important information.

To view the settings for a JavaScript Action:

1. Navigate to the **JavaScript Action** form.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Click to expand **User Interface**.
   c. Select **Menu Items** .

d. Double-click the row representing the configuration you want to edit.

e. In the Menu Item form, navigate to the **Menu Item Contexts** tab.

f. Double-click the row representing the configuration you want to edit.

g. Locate the **Menu Item Action** attribute. Click the Lookup icon next to the Action attribute, and click the **Open** icon.

2. For an explanation of the JavaScript Action attributes, see the Basics table.

3. Click **Close**to return to the Menu Item Context form.

   Tip: You can change the Required Role setting here.

4. Click **Close** to return to the Menu Item form.

   Tip: You can change the □ Enabled attribute here.

Java Script Basics

Attribute	Description
Name	The name that NNMi assigned to this menu item.
JavaScript	The actual JavaScript code that NNMi provided for this menu item

Configure Java Actions

NNMi provides a set of menu items implemented as Java classes. These menu items launch a new browser window.

Do not make any changes to these other than:

- Hide the menu item from the NNMi console (see the □ Enabled attribute "Configure Menu Item Basic Details" on page 1417).
- Change the Required Role setting (see "Configure Menu Item Context Basic Details" on page 1420).
- Change the width / height of the displayed window (see below).
- Enable / disable browser decorations for the displayed browser window (see below).

Caution: If you make changes to a Menu Item provided by NNMi, those changes are at risk of being overwritten in the future. See Author form for important information.

To view the settings for a Java Action:

1. Navigate to the **Java Action** form.

   a. From the workspace navigation panel, select the **Configuration** workspace.

   b. Click to expand **User Interface**.

   c. Select **Menu Items**.

   d. Double-click the row representing the configuration you want to edit.

   e. In the Menu Item form, navigate to the **Menu Item Contexts** tab.

f. Double-click the row representing the configuration you want to edit.

g. Locate the **Menu Item Action** attribute. Click the Lookup icon next to the Action attribute, and click the **Open** icon.

2. For an explanation of the Java Action attributes, see the Basics table.

3. Click **Close** to return to the Menu Item Context form.

   Tip: You can change the Required Role setting here.

4. Click **Close** to return to the Menu Item form.

   Tip: You can change the □ Enabled attribute here.

Java Action Basics

Attribute	Description
Name	The name that NNMi assigned to this menu item.
Java Class	The Java class that NNMi implemented for this menu item
Parameters (Optional)	The list of any parameters used by the menu item.
Browser Width	Indicates the expected behavior of the menu item.
Browser Height	Indicates the expected behavior of the menu item.
Add Browser Decorations	Indicates the expected behavior of the menu item.

Specify Optional Menu Item Enablement Filters

If your SNMP Graph Action or Launch Action applies to Nodes, Interfaces, or Incidents, you can use the Filters Editor to create expressions that further define the context in which this Graph Action or Launch Action is available within NNMi. A Menu Enablement Filter limits the use of the Menu Item which uses this context. The Menu Item is disabled unless the selected object passes this filter.

Design complex Filters on paper as a Boolean expression first to minimize errors when entering your expressions using this Filters editor.

To create any Filter expressions:

1. Navigate to the **Menu Item Context** form.

   a. From the workspace navigation panel, select the **Configuration** workspace.

   b. Click to expand **User Interface**

   c. Select **Menu Items**.

   d. Do one of the following:

      - To create a Menu Item definition, click the New icon.
      - To edit a Menu Item definition, double-click the row representing the configuration you want to edit.

   e. Navigate to the **Menu Item Contexts** tab.

   f. Do one of the following:

      - To create a Context configuration, click the New icon.
      - To edit a Context configuration, double-click the row representing the configuration you want to edit.

2. Navigate to the **Menu Item Enablement Filter** tab.

3. Establish the appropriate settings for the filter you want to create. (See the Custom Filter Editor Components table.)

   When creating any filters, note the following:

   - The Menu Item Enablement filters apply only to Node, Interface, and Incident Object Types. If you select an attribute that is not valid for the Object Type, that part of the filter is not applied.
   - Boolean Attributes begin with "is" and must contain the value `true` or `false`.
   - Each set of expressions associated with a Boolean Operator is treated as if it were enclosed in parentheses and evaluated together. View the expression displayed under **Filter String** to see the logic of the expression as it is created.
   - The AND Boolean Operators must contain at least two expressions.

   - The placement of your cursor and the subsequent text that is selected is important when performing operations using the Additional Filters Editor. For example, you append to or replace, the expression that is selected.
   - The placement of your cursor and the subsequent text that is selected is especially important when adding your Boolean operators. See "Add Boolean Operators in the Additional Filters Editor" on page 310 for more information.
   - You can drag any of the following items to a new location in the Filter String:

     - Filter Editor Options: AND, OR, NOT, EXISTS, NOT EXISTS
     - Filter Expression (Attribute, Operator and Value)

   - When moving items in the Filter String, note the following:
     - Click the item you want to move before dragging it to a new location.
     - As you drag a selected item, an underline indicates the target location.
     - If you are moving the selection up, NNMi places the item above the target location.
     - If you are moving the selection down, NNMi places the item below the target location.
     - If you attempt to move the selection to an invalid target location, NNMi displays an error message.

4. Click **Save and Close** to save and apply your changes.

Custom Filter Editor Components

Attribute	Description
Attribute	The attribute name NNMi should use as the filter criteria. Possible attributes include the following: **Note**: Boolean Attributes begins with "is" and must contain the value `true` or `false`. Interface [click here for a list of attribute values] **Unique Keys from the Interface Form: Capabilities Tab**: - capability (Unique Key of the Capability) **Values from the Interface Form: Custom Attributes Tab**: - customAttrName (Custom Attribute Name) - customAttrValue (Custom Attribute Value) Node [click here for a list of attribute values] **Values from the Basics information on the Node Form:** - isSnmpNode (Agent Enabled) - isSnmpInterface (Agent Enabled) - isNnmSystemLocal (NNMi Management Server) **Values from the Node Form: General Tab**: - sysOidNode (System Object ID) - sysOidInterface (System Object ID) **Unique Keys from the Node Form: Capabilities Tab**: - capability (Unique Key of the Capability) **Values from the Node Form: Custom Attributes Tab**: - customAttrName (Custom Attribute Name) - customAttrValue (Custom Attribute Value) **Values from the Basics information on the Device Profile Form:** - devVendorNode (Device Vendor) - devFamilyNode (Device Family) - devVendorInterface (Device Vendor) - devFamilyInterface (Device Family) Incident [click here for a list of attribute values] **Values from the Incident Form: Custom Attributes Tab**: - customAttrName (Custom Attribute Name)

Custom Filter Editor Components, continued

Attribute	Description
	• customAttrValue (Custom Attribute Value)
Operator	The standard query language (SQL) operations to be used for the search. Valid operators are described below. **Note**: Only the `is null` Operator returns null values in its search. • **=** Finds all values equal to the value specified. • **!=** Finds all values not equal to the value specified. • **<** Finds all values less than the value specified. • **<=** Finds all values less than or equal to the value specified. • **>** Finds all values greater than the value specified. • **>=** Finds all values greater than or equal to the value specified. • **between** Finds all values equal to and between the two values specified. • **in** Searches for a match in at least one of a series of values. • **is not null** Searches for all non-blank values. • **is null** Searches for all blank values. • **like** Enables you to find matches using the asterisk (*) and question mark (?) as wildcard characters. Question mark character means "any single character of any type at this location". Asterisk character means "any number of characters of any type at this location". • **not between** Finds all values except those between the two values specified. • **not in** Finds all values except those included in the list of values. • **not like** Finds all values except those included in the value specified. The not like operator enables you to use the asterisk (*) and question mark (?) as wildcard characters.
Value	The value for which you want NNMi to search. Note the following: • The values you enter are case sensitive. **Note**: When entering the Boolean values, `true` or `false`, use all lowercase. • NNMi displays a variable number of value fields depending on the Operator selected. For example, the `between` Operator causes two value fields to be displayed. • The `in` and `not in` operators require that each value be entered on a separate line.

Additional Filters Editor Buttons

Button	Description
Append	Appends the current expression (Attribute, Operator,and Value) to the selected expression already included in the filter string.
Insert	Inserts the current expression (Attribute, Operator,and Value) in front of the cursor location within the Filter String.
Replace	Replaces the selected expression with the expression displayed in the Attribute, Operator, and Value fields.
AND	Inserts the AND Boolean Operator in the selected cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
OR	Inserts the OR Boolean Operator in the current cursor location. **Note**: View the expression displayed under **Filter String** to see the logic of the expression as it is created.
NOT	Can be used in any part of the Filter String to specify that NNMi should exclude nodes with values that pass the expression that immediately follows the NOT. For example, when evaluating the following Filter String, NNMi includes all nodes that have SNMP enabled and excludes any nodes with a Device Profile attribute value that includes **Cisco** as the Vendor value: `(isSysName = true AND NOT (devVendorNode=Cisco))`
EXISTS	Used for filters that include Capabilities or Custom Attribute names and values in the Filer String. Indicates that you want NNMi to consider nodes that do not have any Capabilities or Custom Attributes when evaluating the Filter String. For example, when evaluating the following Filter String, NNMi includes all nodes with a Capability having the Unique Value of **com.hp.nnm.capability.metric.cse** and ImportantRouters value of **Building5**: `(capability = com.hp.nnm.capability.card.cisco.c2900 AND EXISTS (customAttrName=ImportantRouters AND customAttrValue=Building5))`
NOT EXISTS	Used for filters that include Capabilities or Custom Attribute names and values in the Filer String. Indicates that you want NNMi to consider interfaces that do not have any Capabilities or Custom Attributes when evaluating the Filter String, but exclude the objects that match the expression that follows the NOT EXISTS. For example, when evaluating the following Filter String, NNMi includes all nodes with a hostname that includes **router**, followed by any number of characters, followed by **hp.com** and excludes any nodes with a Custom Attribute named **ImportantRouters** with the value of **Building5**: `(hostname like router*.hp.com AND NOT EXISTS (customAttrName=ImportantRouters AND customAttrValue=Building5))`

Additional Filters Editor Buttons, continued

Button	Description
Delete	Deletes the selected expression. **Note**: If the Boolean Operator is selected, the Additional Filters Editor deletes all expressions associated with the Boolean Operator.

Managing MIBs

NNMi uses MIB information to assist in monitoring the health of your network objects. NNMi also enables you to use MIB Expressions to specify additional information that NNMi should poll. (See "Configure MIB Expressions" on page 1473 for more information.)

To manage these MIBs, perform any of the following tasks:

Tip: Often, NNMi enables you to access the same MIB information in multiple ways. See the help topic for each of the following tasks to determine the options that best meet your needs.

- "View the MIBs Loaded on the NNMi Management Server" on the next page
- Examine MIB Variables
- "Determine the MIBs Supported for a Node (for Administrators)" on the next page
- "Display a MIB Table (MIB Browser)" on page 1453
- "Determine the MIB Variables Supported for a Node (for Administrators)" on page 1454
- "Display a MIB File's Contents (Administrators)" on page 1456
- "Upload MIB Files from the Console" on page 1457
- Load MIBs from the Console or Command Line
- "Unload MIBs " on page 1472
- "Configure MIB Expressions" on page 1473
- "Override MIB OID Types" on page 1484

Examine Available MIBs and MIB Variables

NNMi enables you to take a proactive approach to network management by using MIB Expressions to specify additional information that NNMi should poll. See "Configure MIB Expressions" on page 1473 for more information.

Note: The MIB files that define the MIB variables included in the MIB Expression that you want NNMi to poll must be loaded on the NNMi management server.

Before you create your MIB Expressions, examine the available MIBs and MIB variables using the following methods:

- "Loaded MIBs View" on the next page
- MIB Browser

View the MIBs Loaded on the NNMi Management Server

To view the MIBs stored in the NNMi Database, do either of the following:

Use the "Loaded MIBs View" below

Use the nnmloadmib.ovpl command. Also see"Load MIBs from the Command Line" on page 1461 for more information.

After a MIB is loaded from the NNMi management server, you can also view the MIB when creating a Custom Poller Collection. See "Create a Policy" on page 449 for more information.

Loaded MIBs View

Use the **Loaded MIBs** option of the **Configuration** workspace to determine the MIBs loaded on the NNMi management server.

Note: The MIB containing a variable you want to use in a MIB Expression must be loaded on the NNMi management server.

To view the MIBs Loaded on the NNMi management server:

1. Navigate to the **Configuration** workspace.
2. Expand **MIBs**.
3. Select **Loaded MIBs**.

   NNMi displays the Name of the MIB and the relative MIB file name for each of the MIBs available.

See "Load MIBs" on page 1458 for information about how to load MIBs.

See "Unload MIBs " on page 1472 and nnmloadmibs.ovpl for information about how to unload MIBs.

Determine the MIBs Supported for a Node (for Administrators)

Tip: See MIB Browser Keyboard Navigation for a description of the keyboard navigation you can use in the MIB Browser.

To view the MIBs (Management Information Base) supported by a selected Node, use the **Tools** → **List Supported MIBs** option from the MIB Browser. This option is useful when configuring MIB Expressions so that you can determine the MIBs and associated MIB variables available for use. It can also help you to determine what additional MIBs you might want to load on the NNMi management server. See "Loaded MIBs View" above for more information.

Note: You can also select a Node or Incident from an Inventory view and use the **Actions**→**List Supported MIBs** option to view the MIBs supported for a Node without accessing the MIB Browser.

Tip: You can right-click any object in a table or map view to access the **Actions** menu.

To view the MIBs supported for a Node from the MIB Browser:

Note: Users can view MIB variable information for those nodes to which they have access or for which they provide a valid community string.

1. Do one of the following:
   - Select **Tools** → **MIB Browser**.
   - Open a MIB Variable form from the Loaded MIBs view and select **Actions** → **MIB Information** → **Browse MIB**.

     Note: You can also access the MIB Browser from a Node or Incident view or form. See Determine a Node's MIB Variable Values (MIB Browser) for more information.

   NNMi displays the MIB Browser.
2. In the **Node** attribute, enter the Node Name or IP address of the Node for which you want to view the MIB Variable values.
3. *Optional*. In the **Community String** attribute, do one of the following:
   - Leave this attribute value blank. NNMi uses the Communication parameters currently configured in the NNMi database for the specified Node (if any).
   - Enter a valid *read community string* for the Node. NNMi uses default SNMP version, timeout, maximum retries, and port parameters provided by the NNMi administrator within the `nms-ui.properties` file.

   For more information, see the "Maintaining NNMi" chapter in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.
4. Select **Tools** → **List Supported MIBs**.

   NNMi displays the textual representation of the OID (Object Identifier) for each MIB that is supported by the Node's SNMP Agent. NNMi also lists any MIB tables that reside in each MIB. When displaying the list, NNMi indicates the MIBs that are supported, but not loaded.

   To access the MIB form for a supported MIB, click the MIB name, for example `ENTITY-MIB`.

To view the MIBs supported for a Node without accessing the MIB Browser:

1. Do one of the following:
   - Select a Node from an Inventory view.
   - Select an Incident from an Incident view.
   - Open a Node or Incident form.

     Note: NNMi uses the Incident's Source Node as the selected Node.
2. Select **Actions** → **MIB Information** → **List Supported MIBs**.

   NNMi displays the textual representation of the OID (Object Identifier) for each MIB that is supported by the Node's SNMP Agent. NNMi also lists any MIB tables that reside in each MIB. When displaying the list, NNMi indicates the MIBs that are supported, but not loaded.

   To access the MIB form for a supported MIB, click the MIB name, for example `ENTITY-MIB`.

Related Topics

"Display a MIB Table (MIB Browser)"

"Display a MIB File's Contents (Administrators)" on page 1456

"Determine the MIB Variables Supported for a Node (for Administrators)" on the next page

Check SNMP Support for a Node (MIB Browser)

Find an Entry in the MIB Browser Output

Export MIB Browser Output

Copy MIB Browser Output (MIB Browser)

Print SNMP MIB Browser Output (MIB Browser)

Display a MIB Table (MIB Browser)

To view the MIB table for a selected MIB variable, use the **Tools** → **MIB Table** menu option from the SNMP MIB Browser. This option is useful for determining all of the attributes and associated values for each instance of the MIB variable in a MIB table.

To view MIB table information for a selected MIB variable:

Note: Users can view MIB variable information for those nodes to which they have access or for which they provide a valid community string.

1. Access the SNMP MIB Browser.

   Do one of the following:

   - Select **Tools** → **MIB Browser**.
   - Open a MIB variable form from the Loaded MIBs view and select **Actions** → **MIB Information** → **Browse MIB**.

   > **Tip:** You can right-click any object in a table or map view to access the **Actions** menu.

   Note: You can also access the SNMP MIB Browser from a node or incident view or form. See Determine MIB Variable Values for more information.

   NNMi displays the MIB Browser.
2. In the **Node** attribute, enter the Node Name or IP address of the Node for which you want to view the MIB Variable values.
3. *Optional*. In the **Community String** attribute, do one of the following:

   - Leave this attribute value blank. NNMi uses the Communication parameters currently configured in the NNMi database for the specified Node (if any).
   - Enter a valid *read community string* for the Node. NNMi uses default SNMP version, timeout, maximum retries, and port parameters provided by the NNMi administrator within the `nms-ui.properties` file.

For more information, see the "Maintaining NNMi" chapter in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.

4. In the **OID** attribute, enter the textual or numeric representation of the Object Identifier for the MIB variable to be used as a starting point for viewing the MIB variable values supported on the specified node.

   Note the following:

   - If you accessed the MIB Browser from a MIB variable form, NNMi provides the OID attribute value using the selected MIB variable.
   - You can obtain a MIB variable OID value using the **Loaded MIBs** view. See "Loaded MIBs View" on page 1451 for more information.

5. Click **Walk**.

   NNMi displays the numeric representation of the OID (Object Identifier) for the MIB variable as well as its associated value.

6. Select the MIB variable of interest.

   Note: The MIB variable must have multiple instances. For example: `interfaces.ifTable.ifEntry.ifIndex.1`

7. Select **Tools** → **MIB Table**.

   NNMi displays the MIB table that is associated with the selected MIB variable. The MIB table includes all of the attributes and associated values for each instance in the MIB table.

Related Topics

Display a MIB File's Contents (SNMP MIB Browser)

"Determine the MIBs Supported for a Node (for Administrators)" on page 1451

"Determine the MIB Variables Supported for a Node (for Administrators)" below

Determine the MIB Variables Supported for a Node (for Administrators)

To view the MIB Variables supported for a node, use the **Tools** → **MIB Browser** menu option. This option is useful for determining the following:

- What is possible to graph for a specified node. For example, you might want to determine whether a Node supports MIB Variables in the RMON2-MIB so that you can decide whether to configure a Line Graph using one or more of the RMON2-MIB's Variables.
- How often the MIB Variable values change. This information helps to determine whether a Line Graph would be a useful tool for monitoring the MIB Variable's values.
- Determine MIB Variables to use for Custom Polling. See "Create Custom Polling Configurations" on page 419 for more information.

To view the MIB Variables supported for a specified node:

Note: Users can view MIB variable information for those nodes to which they have access or for which they provide a valid community string.

1. Do one of the following:
   - Select **Tools** → **MIB Browser**.
   - Open a MIB Variable form from the Loaded MIBs view and select **Actions** → **MIB Information** →**Browse MIB**.

   > **Tip:** You can right-click any object in a table or map view to access the **Actions** menu.

   Note: You can also access the MIB Browser from a Node or Incident view or form. See Determine a Node's MIB Variable Values for more information.

   NNMi displays the MIB Browser.
2. In the **Node** attribute, enter the Node Name or IP address of the Node for which you want to view the MIB Variable values.
3. *Optional*. In the **Community String** attribute, do one of the following:
   - Leave this attribute value blank. NNMi uses the Communication parameters currently configured in the NNMi database for the specified Node (if any).
   - Enter a valid *read community string* for the Node. NNMi uses default SNMP version, timeout, maximum retries, and port parameters provided by the NNMi administrator within the `nms-ui.properties` file.

   For more information, see the "Maintaining NNMi" chapter in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.
4. If you accessed the MIB Browser from a MIB Variable form, NNMi provides the OID attribute value using the selected MIB Variable. Otherwise, NNMi provides `mib-2.system` (the root of the MIB-2 branch). To change the OID:
   - Type additional numbers or text strings for a specific MIB-2 area.
   - Replace the default OID numbers to issue an SNMP getNext request for another area in the Internet MIB tree.

   Click here for more information. Note the following:
   - You can obtain a MIB Variable OID value using the **Loaded MIBs** view. See "Loaded MIBs View" on page 1451 for more information.
   - The OID must begin with a dot (.).
   - NNMi automatically completes the OID name for you. The name you begin to enter must be one of the following:
     - A valid textual or numeric OID.
     - An OID alias provided by NNMi. To obtain a list of valid OID aliases, use the **Tools** → **OID Aliases** option from the SNMP MIB Browser.
5. Press **Enter**. NNMi does the following:

If the associated MIB file is loaded on the NNMi management server, NNMi displays the textual representation of the OID (Object Identifier) for the MIB Variable as well as its associated value. If the associated MIB file is not loaded on the NNMi management server, NNMi displays the numeric representation of the OID.

Note: You can also click the ▶ Walk button to display MIB Browser output.

6. To expand a MIB or MIB Variable entry, do one of the following:
   - Click the ▶ Expand icon that precedes the entry you want to expand.
   - Click **Expand Selected**.
7. To collapse a MIB or MIB Variable entry, do one of the following:
   - Click the ▼ Collapse icon that precedes the entry you want to collapse.
   - Click **Collapse Selected**.
8. To stop gathering the MIB Variable information before NNMi reaches the end of the Internet MIB tree, click the ● Stop button.

   When all available MIB Variable values are displayed, the ● Stop button is disabled.

Related Topics

"Display a MIB Table (MIB Browser)"

"Display a MIB File's Contents (Administrators)" below

"Determine the MIBs Supported for a Node (for Administrators)" on page 1451

Check SNMP Support for a Node (MIB Browser)

Find an Entry in the MIB Browser Output

Export SNMP MIB Browser Output

Copy MIB Browser Output (MIB Browser)

Print MIB Browser Output (MIB Browser)

Display a MIB File's Contents (Administrators)

To view a MIB file's contents, use the **Actions** → **Display MIB File** menu option. This option is useful for examining the contents of an entire MIB file to determine all of the MIB Variables and associated values contained in a MIB. You might want to use this option to familiarize yourself with a MIB before creating a MIB expression.

> **Tip:** You can right-click any object in a table or map view to access the **Actions** menu.

To view a MIB file's contents:

1. Do one of the following:
   - Select a MIB from the **Configuration** → **MIBs** → **Loaded MIBs** view.
   - Open a MIB form.

> **Note**: You can also access a MIB form using **Tools** → **List Supported MIBs** from the MIB Browser.

- Open a MIB Variable form.

2. Select **Actions** → **MIB Information** → **Display MIB File**.

   NNMi displays the MIB file's contents.

You can also view a MIB file's contents from the MIB Browser. See Display a MIB File's Contents (MIB Browser) for more information.

Related Topics

"Display a MIB Table (MIB Browser)" on page 1453

"Determine the MIBs Supported for a Node (for Administrators)" on page 1451

"Determine the MIB Variables Supported for a Node (for Administrators)" on page 1454

Check SNMP Support for a Node (MIB Browser)

Find an Entry in the MIB Browser Output

Export MIB Browser Output

Copy MIB Browser Output (MIB Browser)

Print MIB Browser Output (MIB Browser)

Upload MIB Files from the Console

To upload local MIBs files so they are available to load into the NNMi database, use the **Tools** → **Upload Local MIB File** menu. The **Upload Local MIB File** enables you to browse to a vendor's site and upload the specified MIB for subsequent MIB loading.

See "Load MIBs from the Console" on the next page for more information about loading MIBs from the NNMi console.

You can also use the **Tools** → **Load/Unload MIB..** to load any incident configuration associated with the MIB. See "Load SNMP Trap Incident Configurations using the Console" on page 774 for more information.

To upload local MIB files from the NNMi console:

1. Do one of the following:
   a. Navigate to the MIB view or form. For example, select **Configuration** → **MIBs**. → **Loaded MIBs**.
   b. Navigate to the MIB Variable view or form. For example, select **Inventory** → **MIB Variables**.
2. Select **Tools** → **Upload Local MIB File**.
3. Click **Browse** to locate the MIB file you want to upload.
4. Click **Upload** to upload the MIB file to the following directory:

Windows:
`%NnmDataDir%\shared\nnm\user-snmp-mibs`

UNIX:
`/var/opt/OV/shared/nnm/user-snmp-mibs`

5. NNMi displays the following information:
   - The full path to the MIB file.
   - Instructions for loading and listing MIB files.

Load MIBs

NNMi requires that a MIB be loaded on the NNMi management server before you can specify that you want to poll a MIB Expression that includes one of that MIB's variables.

You might also want to load MIBs when creating Graphs.

NNMi automatically stores a set of MIB files on the NNMi management server during installation. These files are located in the following directory:

Windows

`%NnmInstallDir%\misc\nnm\snmp-mibs`

UNIX

`/opt/OV/misc/nnm/snmp-mibs`

To view the MIBs loaded on the NNMi management server, see "View the MIBs Loaded on the NNMi Management Server" on page 1451

To load additional MIBs, do one or more of the following:

- In the console, load MIBs
- "In a command line, load MIBs"

If you are using MIBs with Custom Poller, see "Enable or Disable Custom Poller" on page 420 and "Create a Custom Poller Collection" on page 421

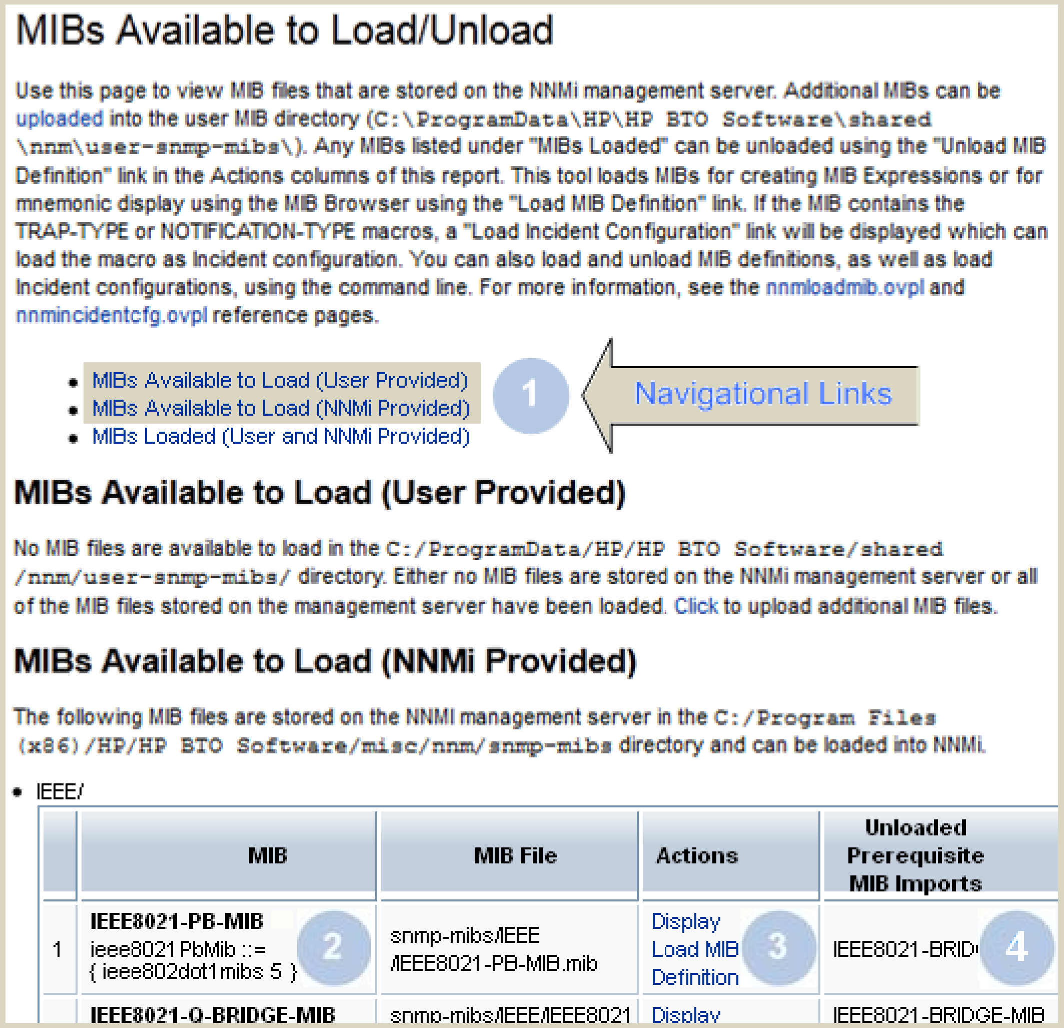
If you are using MIBs to create Graphs, see "Configure SNMP Line Graph Actions" on page 1437

To unload a MIB file, see "Unload MIBs " on page 1472 or use the nnmloadmib.ovpl command.

Load MIBs from the Console

To load additional MIBs from the NNMi console, select the **Configuration** workspace, **MIBs** > **MIB Variables** view. Then, use the **Tools** → **Load/Unload MIB...** menu. The **Load/Unload MIBs** option enables you to view the MIBs that are available to load or unload.

Click here for details.



MIBs Available to Load/Unload

Use this page to view MIB files that are stored on the NNMi management server. Additional MIBs can be uploaded into the user MIB directory (`C:\ProgramData\HP\HP BTO Software\shared \nnm\user-snmp-mibs\`). Any MIBs listed under "MIBs Loaded" can be unloaded using the "Unload MIB Definition" link in the Actions columns of this report. This tool loads MIBs for creating MIB Expressions or for mnemonic display using the MIB Browser using the "Load MIB Definition" link. If the MIB contains the TRAP-TYPE or NOTIFICATION-TYPE macros, a "Load Incident Configuration" link will be displayed which can load the macro as Incident configuration. You can also load and unload MIB definitions, as well as load Incident configurations, using the command line. For more information, see the nnmloadmib.ovpl and nnmincidentcfg.ovpl reference pages.

- MIBs Available to Load (User Provided)
- MIBs Available to Load (NNMi Provided)
- MIBs Loaded (User and NNMi Provided)

MIBs Available to Load (User Provided)

No MIB files are available to load in the `C:/ProgramData/HP/HP BTO Software/shared /nnm/user-snmp-mibs/` directory. Either no MIB files are stored on the NNMi management server or all of the MIB files stored on the management server have been loaded. Click to upload additional MIB files.

MIBs Available to Load (NNMi Provided)

The following MIB files are stored on the NNMI management server in the `C:/Program Files (x86)/HP/HP BTO Software/misc/nnm/snmp-mibs` directory and can be loaded into NNMi.

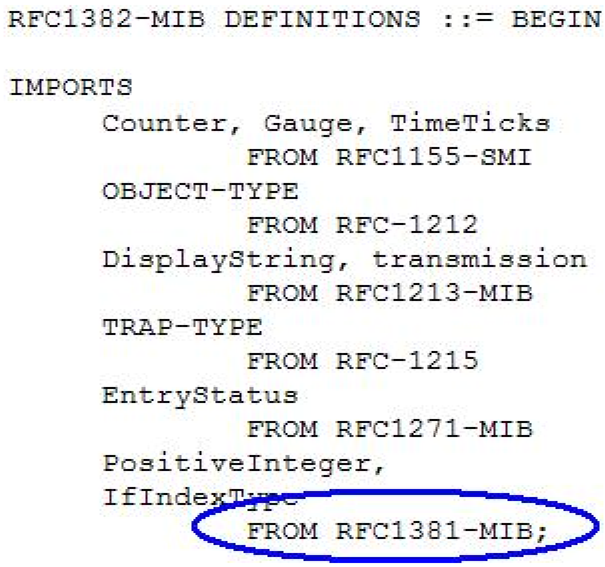
- IEEE/

	MIB	MIB File	Actions	Unloaded Prerequisite MIB Imports
1	**IEEE8021-PB-MIB** ieee8021PbMib ::= { ieee802dot1mibs 5 }	snmp-mibs/IEEE /IEEE8021-PB-MIB.mib	Display Load MIB Definition	IEEE8021-BRID
	IEEE8021-Q-BRIDGE-MIB	snmp-mibs/IEEE/IEEE8021	Display	IEEE8021-BRIDGE-MIB

Load/Unload MIBs Web Page

Feature	Description
1	If any MIBs are stored on the NNMi management server (available for loading or already loaded), click the link to display the appropriate table.
2	If any MIB includes a conforming SNMPv2c SMI *MODULE-IDENTITY*, a text string displays that describes the MODULE-IDENTITY.
3	Select **Load MIB Definition** to load the selected MIB.
4	This column displays any MIBs that are "prerequisites" for the listed MIB, and still need to be manually loaded before you can load the listed MIB. These dependencies are gathered from the MIB's IMPORTS statement. For example:

Load/Unload MIBs Web Page, continued



```
RFC1382-MIB DEFINITIONS ::= BEGIN

IMPORTS
        Counter, Gauge, TimeTicks
                FROM RFC1155-SMI
        OBJECT-TYPE
                FROM RFC-1212
        DisplayString, transmission
                FROM RFC1213-MIB
        TRAP-TYPE
                FROM RFC-1215
        EntryStatus
                FROM RFC1271-MIB
        PositiveInteger,
        IfIndexType
                FROM RFC1381-MIB;
```

Note: Currently, the NNMi console does not display any `TEXTUAL-CONVENTION` entries from loaded MIBs.

You can also use the **Tools** > **Load/Unload MIB...** to load any incident configuration associated with the MIB. See "Load SNMP Trap Incident Configurations using the Console" on page 774 for more information.

To load additional MIBs from the NNMi console:

1. Do one of the following:

   a. Navigate to the MIB view or form. For example, Select **Configuration** > **MIBs** > **Loaded MIBs**.

   b. Navigate to the MIB Variable view or form. For example, Select **Inventory** > **MIB Variables**.

2. Select **Tools** > **Load/Unload MIB...**.

   NNMi displays the following information:

   - MIBs (User provided) that are stored on the NNMi management server and that were provided by the NNMi administrator.
   - MIBs (NNMi provided) that NNMi has stored on the NNMi management server during installation.
   - MIBs that are loaded in the NNMi database.

     See Click here for more details for more information.

3. Navigate to the Unloaded MIB view of interest. For example, **MIBs Available to Load (NNMi Provided)**.

4. In the **MIB** column, find the MIB you want to load. For example, **RFC1381-MIB**.

5. To view the MIB before loading, in the **Actions** column, click **Display**.

NNMi displays the MIB file contents.

6. To load the MIB, in the Actions column, click **Load MIB Definition**.

   NNMi displays the MIB File load progress including the following:

   - The MIB root object identification (OID) number
   - Number of MIBs, MIB variables, enumerated values, table indices, and parent/child hierarchies created
   - Whether the MIB successfully loaded

Also see the nnmloadmib.ovpl command.

To upload a local MIB file so that it is stored on the NNMi management server and available for loading, see "Upload MIB Files from the Console" on page 1457.

To unload a MIB file, see "Unload MIBs " on page 1472.

Load MIBs from the Command Line

To load additional MIBs from the command line, use the nnmloadmib.ovpl command.

> **Note:** You can also use the nnmloadmib.ovpl command with the `-list` option to view the list of MIBs stored in the NNMi database.

To load additional MIBs from the command line:

1. Locate the MIB file you want to use.

   > **Note:** You can use the device vendor's website to locate the MIBs available for your devices.

2. Copy the MIB file to the location of your choice. In the example used in the next step, the MIB file is copied to a `/temp` directory.

3. Use the nnmloadmib.ovpl command to load the MIB on the NNMi management server.

   For example, to load the HOST-RESOURCES-MIB that was copied to the `/temp` directory, you would enter a command similar to the following:

   If you do not want to enter an NNMi User Name attribute value and an NNMi Password attribute value at the command line, you can use the nnmsetcmduserpw.ovpl command to specify the valid user name and password (instead of `-u` and `-p`). The credentials set using the `nnmsetcmduserpw.ovpl` command are valid for command execution by the same user. See "Set Up Command Line Access to NNMi" on page 577 for more information.

   ```
   nnmloadmib.ovpl -u <NNMiadminUsername> -p <NNMiadminPassword> -load
   /temp/HostResources.mib
   ```

If you are using MIBs to create MIB Expressions for Custom Poller, also see "Enable or Disable Custom Poller" on page 420and "Create a Custom Poller Collection" on page 421

If you are using MIBs to create Graphs, see "Configure SNMP Line Graph Actions" on page 1437

To unload a MIB, see "Unload MIBs " on page 1472 or use the nnmloadmib.ovpl command.

Loaded MIBs Form

The MIB form provides details about the selected MIB that is loaded on the NNMi management server.

For information about each tab:

MIB Basics Attributes

Attribute	Description
Name	Name from the DEFINITIONS clause in the MIB file.
MIB File	Relative location of the MIB file.

.

MIB Variable Form (for Administrators)

The MIB Variable form enables you to view more detailed information about the MIB variables available from a MIB that is loaded on the NNMi management server.

For information about each tab:

To view MIB variable information for a MIB that is loaded on the NNMi management server:

1. From the workspace navigation panel, select the **Configuration** workspace.
2. Expand **MIBs**.
3. Select **Loaded MIBs**.
4. Double-click the row representing the MIB.
5. Navigate to the **MIB Variables** tab.
6. Double-click the row representing the MIB Variable.
7. View the Basic attributes (see the MIB Variable Basic Attributes table)

MIB Variable Basic Attributes

Attribute	Description

Name	The Name value that is stored in the MIB definition for the selected MIB variable. In the following example, `ifAdminStatus` is the Name of the MIB variable :
	`ifAdminStatus` `OBJECT-TYPE`
	`SYNTAX INTEGER {`
	`up(1), -- ready to pass packets`
	`down(2),`
	`testing(3) -- in some test mode`
	` }`
	` ACCESS read-write`
	` STATUS mandatory`

MIB Variable Basic Attributes, continued

Attribute	Description

	`DESCRIPTION`
	`"The desired state of the interface. The testing(3) state indicates that no operational packets can be passed."`
	`::= { ifEntry 7 }`

OID (Numeric)	The numeric representation of the OID (Object Identification) value for the selected MIB variable.
OID (Text)	The textual representation of the OID for the selected MIB variable.
Syntax	The SYTNAX value for the MIB variable. Valid values for MIB variable that can be included in a MIB Expression include the following: • Agent Capabilities (SNMP v2 only) • Bits • Counter • Counter64 • Display String (SNMP v1 only) • Enumeration • Gauge • Integer • IP Address • MIB Defined (Indicates a custom type defined in the MIB) • Module Compliance (SNMP v2 only) • Module Identity (SNMP v2 only) • Notification Group (SNMP v2 only) • Notification Type (SNMP v2 only) • Object Group (SNMP v2 only) • Object Identifier • Object Identity (SNMP v2 only) • Octet String • Opaque • Other (Usually indicates the SYNTAX is unset) • Physical Address (SNMP v1 only) • Sequence

MIB Variable Basic Attributes, continued

Attribute	Description
	• Sequence of • Textual Convention (SNMP v2 only) • Time_Ticks • Unsigned32 (Integer) For more information, click here. • When evaluating MIB expressions that include MIB variables of type Counter, Counter64, or Time_Ticks, NNMi evaluates the MIB Variable using the difference in value between the most recent poll and the poll before it. If you want NNMi to calculate a rate over time in seconds, divide the MIB Expression by sysUptime. For example: `(((ifInOctets+ifOutOctets)*8/ifSpeed)*100)/sysUpTime*0.01` **Tip**: The sysUpTime variable is a value of hundredths of a second. When you want the rate in seconds, use `sysUpTime*0.01` in the MIB expression as shown in the previous example. • If you use a MIB variable of type Counter, Counter64, or Time_Ticks in the MIB Expression, NNMi automatically collects sysUpTime values if sysUpTime is not already in the MIB Expression. NNMi uses the sysUptime value to detect a system reboot. Any time a system reboot is detected, NNMi cannot determine the difference in values between polls for any Counter MIB variable and therefore does not calculate the MIB Expression for that poll.
Textual Convention	Defines the format rules to be used when displaying the MIB value. See "MIB Textual Conventions Form" on page 1470for more information.
MIB	The name value that is stored at the beginning of the MIB definitions to identify the MIB. In the following example, RFC1213-MIB is the name of the MIB: `RFC1213-MIB DEFINITIONS ::= BEGIN`

Description	The Description that is stored in the MIB for the selected MIB variable. The following example includes the description for `ifAdminStatus` in the `RFC1213-MIB`:
	`ifAdminStatus OBJECT-TYPE`
	` SYNTAX INTEGER {`
	` up(1), -- ready to pass packets`
	` down(2),`
	` testing(3) -- in some test mode`
	` }`
	`ACCESS read-write`
	` STATUS mandatory`
	` DESCRIPTION`
	` **"The desired state of the interface. The testing(3) state indicates that no operational packets can be passed."**`
	`::= { ifEntry 7 }`

Enumerated Values Form (for Administrators)

The Enumerated Values form enables you to view each enumerated value pair, if any, for a selected MIB variable. For example, the `ifAdminStatus` MIB variable, includes enumerated values for status as shown in the following example:

```
ifAdminStatus OBJECT-TYPE
SYNTAX INTEGER {
1 up,
2down),
3 testing
 }
 ACCESS read-write
 STATUS mandatory
 DESCRIPTION
"The desired state of the interface. The testing(3) state
 indicates that no operational packets can be passed."
 ::= { ifEntry 7 }
```

The enumerated values are included in the following table:

Enumerated Values for ifAdminStatus

String Value	Numeric Value
up	1
down	2
testing	3

For information about each tab:

To view the enumerated values for a selected MIB variable:

1. From the workspace navigation panel, select the **Configuration** workspace.
2. Expand **MIBs**.
3. Select **Loaded MIBs**.
4. Double-click the row representing the MIB.
5. Select the **MIB Variables** tab.
6. Double-click the row representing the MIB Variable.
7. Select the **Enumerated Values** tab.

   NNMi displays the string and numeric value for each enumeration, if any, specified for the selected MIB variable.
8. To view more details about an enumerated value pair, double-click the row representing the value pair.
9. View the Basics information for the selected Enumerated Value (see the Enumerated Value Basic Attributes table).

Enumerated Value Basic Attributes

Attribute	Description
String Value	The text value that is associated with the Numeric Value for the selected MIB variable.
Numeric Value	The numeric value that is associated with the String Value for the selected MIB variable.
MIB Variable	The name of the selected MIB variable that contains enumerated values. For example, `ifAdminStatus` is a MIB Variable that contains enumerated values.
MIB	The name value that is stored at the beginning of the MIB definitions to identify the MIB. In the following example, `RFC1213-MIB` is the name of the MIB: `RFC1213-MIB DEFINITIONS ::= BEGIN`

Table Indices Form (for Administrators)

The Table Index form enables you to view the index values, if any, for a selected MIB variable. Table indices are identified using the INDEX keyword as shown in the following example for the `atEntry` MIB variable:

```
atEntry OBJECT-TYPE
 SYNTAX AtEntry
 ACCESS not-accessible
 STATUS deprecated
 DESCRIPTION
 "Each entry contains one NetworkAddress to
 `physical' address equivalence."
 INDEX { atIfIndex,
atNetAddress }
 ::= { atTable 1 }
```

In the example, `atIfIndex` and `atNetAddress` are table indices for the `atEntry` MIB variable.

Table indices are used to store multiple values for a single MIB variable.

For information about each tab:

To view the table index values for a selected MIB variable:

1. From the workspace navigation panel, select the **Configuration** workspace.
2. Expand **MIBs**.
3. Select **Loaded MIBs**.
4. Double-click the row representing the MIB.
5. Select the **MIB Variables** tab.
6. Double-click the row representing the MIB Variable.
7. Select the **Table Indices** tab.

NNMi displays the Position and Name for each of the Table Indices, if any, specified for the selected MIB variable.

8. To view more details about a specific Table Index entry, double-click the row representing the Table Index entry.
9. View the Basics information for the selected Table Index (see the Table Index Basic Attributes table).

Table Index Basic Attributes

Attribute	Description

Position	The position number of the MIB variable that is used as a Table Index object. In the following example, `atIfIndex` and `atNetAddress` are MIB Variables used as Table Index objects. `atIfIndex` is position 0 and `atNetAddress` is position 1:
	`INDEX {` **`atIfIndex,`**
	`atNetAddress` `}`

MIB Variable	The name of the selected MIB variable that is used as a Table Index object. Table indices are used for storing multiple values for a MIB variable.

Table Definition	The name of the MIB variable used to define the MIB table. In the following example, `atEntry` is the MIB variable that defines the MIB table:
	`atEntry` `OBJECT-TYPE`
	`SYNTAX AtEntry`
	`ACCESS not-accessible`
	`STATUS deprecated`
	`DESCRIPTION`
	`"Each entry contains one NetworkAddress to`
	`` `physical' address equivalence." ``
	`INDEX { atIfIndex,`
	`atNetAddress }`
	`::= { atTable 1 }`

MIB Name	The name value that is stored at the beginning of the MIB definitions to identify the MIB. In the following example, RFC1213-MIB is the name of the MIB: `RFC1213-MIB DEFINITIONS ::= BEGIN`

MIB Notification Form (for Adminstrators)

The MIB Notification form enables you to view the SNMP trap information, if any, that is defined by the selected MIB.

For information about each tab:

To view the MIB Notification information for a selected MIB:

1. From the workspace navigation panel, select the **Configuration** workspace.
2. Expand **MIBs**.

3. Select **Loaded MIBs**.
4. Double-click the row of interest.
5. Select the **MIB Notifications** tab.
6. View the Basics information for the selected MIB Notification (see the MIB Notification Basic Attributes table).

MIB Notification Basic Attributes

Attribute	Description

Name	The Name value that is stored in the MIB definition for the selected MIB notification. In the following example, `linkDown` is the Name of the MIB variable :
	`linkDown` `NOTIFICATION-TYPE`
	` OBJECTS {` **`ifIndex, ifAdminStatus, ifOperStatus`** `}`
	` STATUS current`
	` DESCRIPTION`
	` "A linkDown trap signifies that the SNMP entity, acting in`
	
	` an agent role, has detected that the ifOperStatus object`
	`for`
	` one of its communication links is about to enter the down`
	` state from some other state (but not from the notPresent`
	` state). This other state is indicated by the included`
	`value`
	` of ifOperStatus."`
	` ::= { snmpTraps 3 }`

OID (Numeric)	The numeric representation of the OID (Object Identification) value for the selected MIB notification.
OID (Text)	The textual representation of the OID for the selected MIB variable.
MIB	The name value that is stored at the beginning of the MIB definitions to identify the MIB. In the following example, `IF-MIB` is the name of the MIB: `IF-MIB DEFINITIONS ::= BEGIN`
Description	SNMP Trap Description that is stored in the MIB.
Type	*Optional*. SNMP Trap --#TYPE value that is stored in the MIB.
Summary	*Optional*. The-- #SUMMARY value that is stored in the MIB for the SNMP Trap.
Arguments	*Optional*. Number of arguments for the SNMP Trap.
Severity	*Optional*. The --#SEVERITY value that is stored in the MIB for the SNMP Trap.
Generic	*Optional*. The --#GENERIC value that is stored in the MIB for the SNMP Trap.
Category	*Optional*. The --#CATEGORY value that is stored in the MIB for the SNMP Trap.
Source ID	*Optional*. The --#SOURCE ID value that is stored in the MIB for the SNMP Trap.
State	*Optional*. The --#STATE value that is stored in the MIB for the SNMP Trap.

Notification Variables Form (for Administrators)

The Notification Variables form enables you to view the SNMP trap information, if any, that can be sent by the selected MIB variable. In the following example `linkDown` is the MIB variable that defines the SNMP trap. `ifIindex`, `ifAdminStatus`, and `ifOperStatus` are the MIB variables that have information that will be included in the SNMP trap:

```
linkDown NOTIFICATION-TYPE
 OBJECTS { ifIndex, ifAdminStatus, ifOperStatus }
 STATUS current
 DESCRIPTION
 "A linkDown trap signifies that the SNMP entity, acting in
 an agent role, has detected that the ifOperStatus object for
 one of its communication links is about to enter the down
 state from some other state (but not from the notPresent
 state). This other state is indicated by the included value
 of ifOperStatus."
 ::= { snmpTraps 3 }
```

For information about each tab:

To view the Notification Variable information for a selected MIB variable:

1. From the workspace navigation panel, select the **Configuration** workspace.
2. Expand **MIBs**.
3. Select **Loaded MIBs**.
4. Double-click the row representing the MIB.
5. Select the **MIB Notifications** tab.
6. Double-click the row representing the MIB Notification.
7. Navigate to the **Notification Variables** tab.
8. Double-click the row representing the Notification Variable.
9. View the Basics information for the selected MIB Notification (see the Notification Variable Basic Attributes table).

MIB Notification Basic Attributes

Attribute	Description

Position	The position number of the MIB variable that is used as a Notification Variable object. The Notification Variable identifies information that is included in the SNMP trap. In the following example, `atIfIndex`, `ifAdminStatus`, and `ifOperStatus` are Notification Variables. `atIfIndex` is position 1, `ifAdminStatus` is position 2, and `ifOperStatus` is position 3:
	`linkDown` `NOTIFICATION-TYPE`
	`OBJECTS {`**`ifIndex, ifAdminStatus, ifOperStatus`**`}`

MIB Variable	The name of the selected MIB variable that is used as a Notification Variable

MIB Notification Basic Attributes, continued

Attribute	Description

	object. In the following example `ifIndex`, `ifAdminStatus`, and `ifOperStatus` are Notification Variables:
	`linkDown NOTIFICATION-TYPE`
	` OBJECTS {` **`ifIndex`**`,` **`ifAdminStatus`**`,` **`ifOperStatus`** `}`

Trap Definition	The name of the MIB notification used to define the SNMP trap . In the following example, `linkDown` is the MIB notification that describes the SNMP trap definition:
	`linkDown` `NOTIFICATION-TYPE`
	` OBJECTS { ifIndex, ifAdminStatus, ifOperStatus }`
	` STATUS current`
	` DESCRIPTION`
	` "A linkDown trap signifies that the SNMP entity, acting`
	`in`
	` an agent role, has detected that the ifOperStatus object`
	`for`
	` one of its communication links is about to enter the down`
	
	` state from some other state (but not from the notPresent`
	` state). This other state is indicated by the included`
	`value`
	` of ifOperStatus."`
	` ::= { snmpTraps 3 }`

MIB	The name value that is stored at the beginning of the MIB definitions to identify the MIB. In the following example, `IF-MIB` is the name of the MIB: `IF-MIB DEFINITIONS ::= BEGIN`

MIB Textual Conventions Form

The MIB Textual Convention form enables you to view the format rules for the selected Textual Convention that are defined in the MIB. NNMi uses these MIB format rules to determine how to display any associated MIB variable values of type Octet String.

For information about each tab:

To view the format rules for a Textual Convention:

1. From the workspace navigation panel, select the **Configuration** workspace.
2. Expand **MIBs**.
3. Select **Textual Conventions**.

   Note: You can also access Textual Conventions for a Loaded MIB, using the **Loaded MIBs** option that appears under the **MIBs** folder.

4. Double-click the row representing the textual convention.
5. View the Basic attributes (see the Textual Conventions Basic Attributes table)

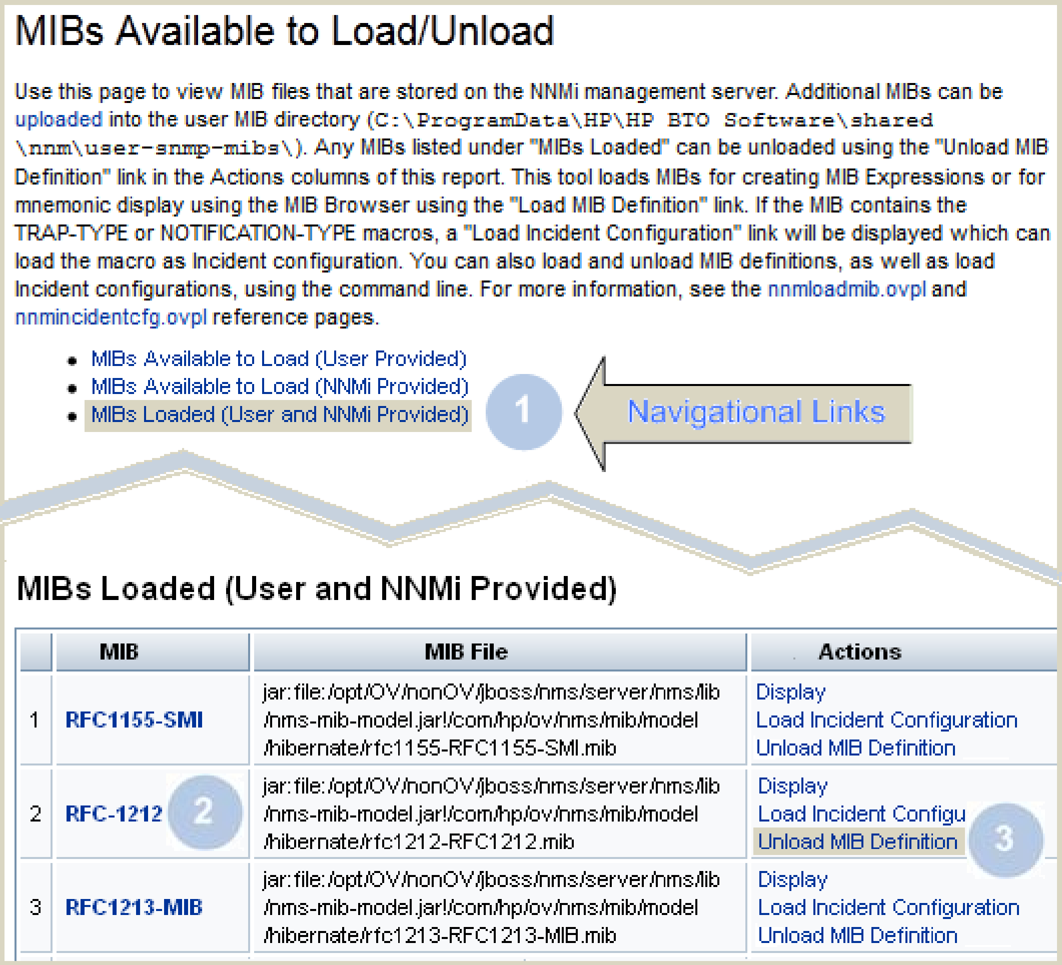
Textual Conventions Basic Attributes

Attribute	Description
Name	The Name value that is stored in the MIB definition for the selected textual convention.
Status	The Status value that is stored in the MIB definition for the selected textual convention. Possible values are: • current • deprecated • obsolete
Display Hint	Format rule used with the Value Constraint and Primitive Type to help determine the format when displaying the associated MIB value. For example, to display the MAC Address, the DISPLAY-HINT is `"1x:"` to indicate the value must consist of a one-byte hex string or two-hex digits, such as `01` or `AB`.
Value Constraint	Format rule used with the Display Hint and Primitive Type to help determine the format when displaying the associated MIB variable value. For example, the value constraint under SYNTAX for the MAC Address is `(SIZE (6))` to indicate the format must include six one-byte hex strings, such as `0A:BC:1D:2E:3F:40`.
Primitive Type	Defines the base type to be used when determining the format for displaying the associated MIB variable value. For example, the MAC Address Primitive Type is `OCTET STRING`. Valid values include the following: • Integer • Unsigned Integer • Octet String • Counter • Counter64 • Gauge • Time_Ticks
MIB	The name value that is stored at the beginning of the MIB definitions to identify the MIB. Click the Lookup icon, and select Open to access the associated MIB that is loaded on the NNMi management server.
Description	The Description that is stored in the MIB for the selected Textual Convention.

Unload MIBs

To unload MIBs from the NNMi console, select the **Configuration** workspace, **MIBs** > **MIB Variables** view. Then, use the **Tools** > **Load/Unload MIB** menu. The **Load/Unload MIB** option also enables you to view the MIBs that are available to load or unload.

Click here for details.



MIBs Available to Load/Unload

Use this page to view MIB files that are stored on the NNMi management server. Additional MIBs can be uploaded into the user MIB directory (`C:\ProgramData\HP\HP BTO Software\shared\nnm\user-snmp-mibs\`). Any MIBs listed under "MIBs Loaded" can be unloaded using the "Unload MIB Definition" link in the Actions columns of this report. This tool loads MIBs for creating MIB Expressions or for mnemonic display using the MIB Browser using the "Load MIB Definition" link. If the MIB contains the TRAP-TYPE or NOTIFICATION-TYPE macros, a "Load Incident Configuration" link will be displayed which can load the macro as Incident configuration. You can also load and unload MIB definitions, as well as load Incident configurations, using the command line. For more information, see the nnmloadmib.ovpl and nnmincidentcfg.ovpl reference pages.

- MIBs Available to Load (User Provided)
- MIBs Available to Load (NNMi Provided)
- MIBs Loaded (User and NNMi Provided)

MIBs Loaded (User and NNMi Provided)

	MIB	MIB File	Actions
1	RFC1155-SMI	jar:file:/opt/OV/nonOV/jboss/nms/server/nms/lib/nms-mib-model.jar!/com/hp/ov/nms/mib/model/hibernate/rfc1155-RFC1155-SMI.mib	Display Load Incident Configuration Unload MIB Definition
2	RFC-1212	jar:file:/opt/OV/nonOV/jboss/nms/server/nms/lib/nms-mib-model.jar!/com/hp/ov/nms/mib/model/hibernate/rfc1212-RFC1212.mib	Display Load Incident Configu Unload MIB Definition
3	RFC1213-MIB	jar:file:/opt/OV/nonOV/jboss/nms/server/nms/lib/nms-mib-model.jar!/com/hp/ov/nms/mib/model/hibernate/rfc1213-RFC1213-MIB.mib	Display Load Incident Configuration Unload MIB Definition

Load/Unload MIBs Web Page

Feature	Description
1	If any MIBs are stored on the NNMi management server (available for loading or already loaded), click the link to display the appropriate table.
2	If any MIB includes a conforming SNMPv2c SMI *MODULE-IDENTITY*, a text string displays that describes the MODULE-IDENTITY. For example, the MODULE-IDENTITY in row 1 is `ianaifType`
3	Select **Unload MIB Definition** to unload the selected MIB.

To unload MIBs from the NNMi console:

1. Do one of the following:

   a. Navigate to the MIB view or form. For example, Select **Configuration** > **MIBs** > **Loaded MIBs**.

   b. Navigate to the MIB Variable view or form. For example, Select **Inventory** > **MIB Variables**.

2. Select **Tools** > **Load/Unload MIB**.

   NNMi displays the following information:

   - MIBs (User provided) that are stored on the NNMi management server and that were provided by the NNMi administrator.

   - MIBs (NNMi provided) that NNMi has stored on the NNMi management server during installation.

   - MIBs that are loaded in the NNMi database.

     See Click here for more details for more information.

3. Navigate to the Loaded MIBs view.

4. In the MIB colum, find the MIB you want to unload.

5. To view the MIB before unloading, in the Actions column, click **Display**.

   NNMi displays the MIB file contents.

6. To unload the MIB, in the Actions column, click **Unload MIB Definition**.

   NNMi displays the MIB File load progress including the following:

   - Name of the MIB definition

   - Whether the unload MIB was successful

Also see the nnmloadmib.ovpl command.

Configure MIB Expressions

NNMi enables you to take a proactive approach to network management by using SNMP MIB Expressions to specify additional information that NNMi should poll. After you create the MIB Expression, you can display this information in Graphs or use it with the NNMi Custom Poller feature.

To specify a MIB Expression, provide the required information within one of the following contexts:

"MIB Expressions Form (Custom Poller)" on page 431

"MIB Expression Form (Line Graph)" on the next page

See "MIB Expressions in Full URLs" on page 1435 for more information about using MIB Expressions in Graphs.

See "Create Custom Polling Configurations" on page 419 for more information about using MIB Expressions with Custom Poller.

The MIB Expressions view in the Configuration workspace includes the MIB Expressions provided by NNMi. See "MIB Expressions View" below for more information.

MIB Expressions View

Use the MIB Expressions view to determine the MIB Expressions available for use. You can use MIB Expressions when configuring Custom Poller and Graph Actions. See and "MIB Expressions Form (Custom Poller)" on page 431 and "MIB Expression Form (Line Graph)" below for more information.

Note: All MIB Expressions provided by NNMi use the Author value **HP Network Node Manager.**

To view the MIB Expressions available:

1. Navigate to the **Configuration** workspace.
2. Select the **MIBs** folder.
3. Select the **MIB Expressions** view.

   The columns in this table view show the Name, Author, and Description for each available MIB Expression.

MIB Expression Form (Line Graph)

You can access the MIB Expression form in the following ways:

- From the **Configuration** workspace > **MIBs**folder > **MIB Expressions** view.
- From the **Configuration** workspace > **Monitoring** folder > **Custom Poller Configuration** form
- From the **MIB Specification** form. (Used when configuring SNMP Graph actions.)

When you want to create a MIB Expression to be used in Line Graphs, use the **MIB Expressions** view. See "Configure MIB Expressions" on the previous page for more information about configuring Line Graph. See "MIB Expressions Form (Custom Poller)" on page 431 for more information about using the **Custom Poller Configuration** form.

> **Note:** You can re-use any MIB Expression that you create for NNMi Line Graphs or for Custom Poller. Use "MIB Expressions View" above to see a list of the available MIB Expressions. Use the "Loaded MIBs View" on page 1451 to see a list of the MIBs loaded on the NNMi management server.

To create a MIB Expression using the MIB Expression form:

1. From the workspace navigation panel, select the **Configuration** workspace.
2. Expand the **MIBs** folder.
3. Select the **MIB Expressions** view.
4. Do one of the following:

- To create a MIB Expression, click the ✱ **New** icon.
- To edit a MIB Expression, double-click the row representing the configuration you want to edit.

5. Provide the required basic settings (see the MIB Expression Basic Attributes table).
6. *Only for Multiple Instance MIB Expressions*. Line Graphs that display multiple instances use the following syntax for the line label that appears in the Graph legend:

   <*node_name*> <*Line_Label*>.<*instance_string*>

   In this instance, <*Line_Label*> is the Line Label value specified when using the MIB Specification for.

   Use the **Instance Display Configuration** section of the MIB Expression form to specify the configuration for the <*instance_string*> values (see the Instance Display Configuration table).

   See "Use the MIB Expression Editor (Line Graph)" on page 1479 for more information about multiple instance MIB Expressions.
7. Click **Save and Close**.
8. To test your MIB Expression, select **Actions** → **Graph MIB Expression**. See "Test a MIB Expression (Line Graph)" on page 1478 for more information.

> **Tip:** You can right-click any object in a table or map view to access the **Actions** menu.
>
> - You must save the MIB Expression before you use **Actions** → **Graph MIB Expression**.
> - The NNMi administrator determines the label that is used to identify the data instances that are displayed in Line Graphs using the Instance Display Configuration (see the Instance Display Configuration table). If the Instance Display Configuration is not set, NNMi identifies each instance that appears in a Line Graph using the Node's short DNS Name followed by the MIB Instance value in the format: <*node_name*> -<*MIB_instance_value*>.

MIB Expression Basic Attributes

Attribute	Description
Unique Key	Used as a unique identifier when exporting and importing MIB Expression definitions. To ensure that the value you enter is unique, it is recommended that you use the Java name space convention when providing this value. It is also useful to include the label value as part of the unique key as shown in the following example: `com.<your_company_name>.nnm.mibexp.<mib_expression_name>` The maximum length is 80 characters. **Note:** Unlike the Unique Key attributes associated with other objects, you can change the MIB Expression configuration's Unique Key value at any time.

MIB Expression Basic Attributes, continued

Attribute	Description
Name	The name you want to use for the MIB information being polled. This name can be the same name as a MIB Variable used in the MIB Expression, or you can enter a name of your choice. Type a maximum of 50 characters. Alpha-numeric and special characters (~ ! @ $ % ^ * ( ) _+) are permitted. No spaces are permitted.
Author	Indicates who created or last modified the MIB Expression. **Caution:** If the Author attribute value is **HP Network Node Manager**, any changes are at risk of being overwritten in the future. - Click **Lookup** and select **Show Analysis** to display details about the currently selected Author. - Click **Quick Find** to access the list of existing Author values. - Click **New** to create an Author value.
Expression	Click the button to access the MIB Expression editor. See "Use the MIB Expression Editor (Line Graph)" on page 1479 for information about using the MIB Expression editor. **Note**: The MIB containing the variable must be loaded on the NNMi management server.
Display numeric MIB OIDs in the Expression	Enables you to display the MIB object identifier (OID) rather than the MIB variable name in the MIB Expression. Select **Display MIB OIDs in the Expression** ☑ to replace any MIB variable name with the MIB OID value in the MIB Expression. Clear **Display MIB OIDs in the Expression** □ to display the MIB variable names rather than the MIB OIDs within the MIB Expression.
Description	NNMi provides the Description attribute to help you further identify the current MIB Expression configuration. Use the description field to provide additional information that you would like to store about the current MIB expression configuration. Type a maximum of 2000 characters. Alpha-numeric and special characters (~ ! @ # $ % ^ & * ( ) _+ -) are permitted.

Instance Display Configuration

Attribute	Description
Conversion Algorithm	Used to determine the format in which the instance portion (<*instance_string*>) of the line label appears in the Line Graph legend.

Instance Display Configuration, continued

Attribute	Description
	Line labels in a Line Graph use the following syntax: <*node_name*> <*Line_Label*>.<*instance_string*> In this instance, <*Line_Label*> is the Line Label value specified when using the MIB Specification form. Possible Conversion Algorithms are: • **Numeric** - Use this option to display the instance number returned by the SNMP query as the <*instance_string*> value. This format is useful when no meaningful name is available in the MIB. For example, Line Graphs that display CPU information might use this format. • **MIB Variable** - Use this option to display the value that is stored in the MIB variable you specify. To obtain each MIB variable value, NNMi appends the instance number to the MIB variable specified. The result from the SNMP query is converted to a text string and displayed as the <*instance_string*> value of the line label in the Line Graph legend. • **Alphabetic** - Use this option to display information for legacy Cisco Arrow Point load balancers. When using this algorithm, each instance number returned by the SNMP query is treated as a set of ASCII characters instead of numbers. For example, the instance 101.120.97.109.112.108.101 would be displayed as 'example' in the <*instance_string*> of the line label.. • **Interface Name** - Use this option to display the interface name as the <*instance_string*> in the Line Graph legend. **Note**: The Interface Name option is only valid when an IfIndex value is returned as the instance number. The ifIndex value is then used to determine the Interface Name value. • **Interface Name Indirect** - Use this option to display the Interface Name value obtained from an indirect reference in the MIB table. For example, if the MIB variable you specify resides in an RMON MIB table, use this algorithm. **Note:** The **Interface Name Indirect** option is only valid when an OID is returned from an SNMP query that, when queried, returns an ifIndex value. The ifIndex value is then used to determine the Interface Name value using the "Interface Name" algorithm.
Display Variable	Select the MIB variable you want to display as the <*instance_string*> value in the line label of the Line Graph legend. NNMi uses the Conversion Algorithm you specify to determine how to obtain the <*instance_string*> value.
Display Filter	When you display the Line Graph, the data displayed in the Line Graph is filtered based on the criteria you provide here. Enter a valid regular expression that specifies the pattern you want NNMi to match when determining the values to display in the <*instance_string*> value of each line

Instance Display Configuration, continued

Attribute	Description
	label. **Note**: NNMi uses the syntax defined for java regular expressions (java.util.regex Pattern class). NNMi finds the first character sequence that matches the Display Filter expression. If NNMi does not find a match for the Display Filter, it returns the Display Variable name. For example, if you have several interfaces with an ifDescr set to "FastEthernet" followed by a unique set of numbers for each interface (such as FastEthernet0/1, FastEthernet0/2, FastEthernet0/3, and so on), you can use the following Display Filter to display "Ethernet" followed by the unique set of numbers: `(Ethernet.*[0-9]+){1}` In the example, the following matches occur: • `Ethernet` matches Ethernet • The `.*` matches 0/ • The `[0-9]+` matches any sequence of numbers • The `{1}` specifies to match the expression exactly one time

Test a MIB Expression (Line Graph)

The Actions menu enables you to test the results of a MIB Expression using a Line Graph.

Note: You must save the MIB Expression before you use **Actions** → **Graph MIB Expression**.

Tip: You can right-click any object in a table or map view to access the **Actions** menu.

To graph the results for a MIB Expression:

1. Navigate to the **MIB Expression** form.
   a. From the workspace navigation panel, select the **Configuration** workspace.
   b. Expand the **MIBs** folder.
   c. Select **MIB Expressions**.

      Note: You can also access the MIB Expression form when creating Line Graphs and when creating Custom Poller Collections. See "MIB Expression Form (Line Graph)" on page 1474 and "MIB Expressions Form (Custom Poller)" on page 431 for more information.
2. Select the row representing the MIB Expression you want to graph.
3. Select **Actions** → **Graph MIB Expression**.

   The dialog for selecting a node appears.

4. Click the **Lookup** icon and select **Quick Find**.
5. Select the node you want to use to test your MIB Expression results.

   NNMi displays a Line Graph using the selected node and calculating the results for the MIB Expression you selected.

   Note the following:

   - *Line Graphs Only*. When evaluating MIB Expressions that include MIB variables of type Counter or Counter64, NNMi requests both the high capacity and low capacity counter variable for any interface instance. If the high capacity Counter 64 is enabled for any given interface instance, NNMi uses the high capacity counter.
   - *Custom Poller Only*. When evaluating MIB Expressions that include MIB variables of type Counter, NNMi requests only the low capacity counter information for any interface instance.

Use the MIB Expression Editor (Line Graph)

Use the MIB Expression Editor to specify the MIB Variables and any Constant values or arithmetic operators you want to include in your MIB Expression.

For example, disk utilization could be calculated and polled using a MIB Expression similar to the following:

(`hrStorageAllocationUnits` * `hrStorageSize`) / (`hrStorageUsed` * `hrStorageAllocation`)

See the MIB Expression Editor Options table for a description of each of the MIB Expression Editor options.

When using the MIB Expression Editor, note the following:

- As a general guideline, begin by writing out the MIB Expression. Then in the MIB Expression Editor, begin creating your MIB Expression by selecting your arithmetic operators (+, -, *, or /) from the outermost parenthesis to the innermost parenthesis. Each time you specify an arithmetic operator (+, -, *, or /), NNMi creates a set of parenthesis to specify the ordering of the mathematical calculation.
- When adding arithmetic operators (+, -, *, or /) to a MIB Expression, first click to select the location in the MIB Expression at which you want to add the arithmetic operator.
- Click to select the arithmetic operator (for example +) in the MIB Expression, before selecting the MIB variable or Constant value that you want to add, subtract, multiply or divide.

  You can also use the following key bindings to add arithmetic operators:

  - ALT+ (plus button)
  - ALT- (minus button)
  - ALT/ (divide button)
  - ALT* (multiply button)
- NNMi inserts arithmetic operators, MIB Expressions, and Constant values from the left to right.

- To replace an arithmetic operator use the <> (Change Operator) button (see table).
- To replace a MIB Variable or Constant value, click to select the existing value in the MIB Expression and then select the new MIB variable or enter the new Constant value.

  Note: You can replace a MIB Variable with another MIB Variable or with a Constant value. You can replace a Constant value with a MIB Variable or Constant value.
- You can drag any of the following items to a new location in the MIB Expression:
  - MIB variable
  - Constant value
  - An operation, such as **(ifInOctets + ifOutOctets)**

Click here for more information about moving items in the MIB Expression to a new location.

When moving items in the MIB Expression, note the following:

- To move an arithmetic operation (for example, **(ifInOctets + ifOutOctets)**), click the arithmetic operator before dragging it to a new location.
- To move a MIB Variable or Constant Value, click the MIB Variable or Constant Value you want to move before dragging it to a new location.
- If you are moving the selected item to the right, NNMi places the item to the right of the new location.
- If you are moving the selected item to the left, NNMi places the item to the left of the new location.
- As you drag a selected item, an underline indicates the current target location.
- If you drag a selected item past the outermost parenthesis, it is deleted. If desired, you can re-enter the value in the new location.

MIB Expression Example

To create a MIB Expression that calculates the percentage of available bandwidth on a half-duplex interface, you might create the following MIB Expression:

((((ifInOctets + ifOutOctets) * 8) / ifSpeed) * 100)

See Creating a MIB Expression Animation for an animated demonstration of creating the MIB Expression above.

Click here for a step-by-step textual example of creating the same MIB Expression:

To create a MIB Expression that calculates the percentage of available bandwidth on a half-duplex interface, you might create the following MIB Expression:

((((ifInOctets + ifOutOctets) * 8) / ifSpeed) * 100)

To create the expression above, begin by specifying each arithmetic operator from the outermost parenthesis to the innermost parenthesis.

1. Click * (multiply).
2. Click / (divide).

Now that you have multiple entries in your MIB Expression, click to select the location in the MIB Expression to which you want to add each remaining arithmetic operators.

3. In the MIB Expression, click / (divide).

   The divide (/) arithmetic operator and its surrounding parenthesis should appear highlighted. Because NNMi inserts arithmetic operators, MIB variables, and Constant values from left to right, selecting / (divide) places the next arithmetic operator to the left of the divide arithmetic operator.

4. Click * (multiply).

   The multiply (*) arithmetic operator and its parenthesis should appear to the left of the divide arithmetic operator you previously selected.

5. In the MIB Expression, click the leftmost * (multiply).

   The multiply (*) arithmetic operator and its surrounding parenthesis should appear highlighted.

6. Click + (add).

   The add (+) arithmetic operator and its parenthesis should appear to the left of the multiply (*) arithmetic operator you previously selected.

   Now that you have specified the arithmetic operators, you are ready to add the MIB variables and Constant values. Begin by selecting the arithmetic operator in the MIB Expression to which you will add MIB variables, Constant values, or both. We will begin with the leftmost arithmetic operation.

   Note: As you add your MIB variables or Constant values, make sure you first select the corresponding arithmetic operator within the MIB Expression.

7. In the MIB Expression attribute, click + (add).

8. Select the ifInOctets MIB Variable:

   a. Click to open the MIB Variable Tree.

   b. Navigate to **ifInOctets**.

   c. Select **ifInOctets**.

   d. Click **OK**.

      The ifInOctets MIB variable should appear to the left of the add (+) arithmetic operator.

9. Select the ifOutOctets MIB Variable:

   a. Click to open the MIB Variable Tree.

   b. Navigate to **ifOutOctets**.

   c. Select **ifOutOctets**.

   d. Click **OK**.

      The ifOutOctets MIB variable should appear to the right of the add (+) arithmetic operator.

You are ready to specify the Constant value 8 that corresponds with the leftmost multiply (*) arithmetic operator.

10. Click the leftmost * multiply.
11. In the Constant attribute, enter 8 and click Enter.

    The value 8 should appear to the right of the multiply (*) arithmetic operator that you previously selected.
12. In the MIB Expression, click divide (/).
13. Select the IfSpeed MIB Variable:
    a. Click to open the MIB Variable Tree.
    b. Navigate to ifSpeed.
    c. Double-click ifSpeed.
    d. Click **OK**.

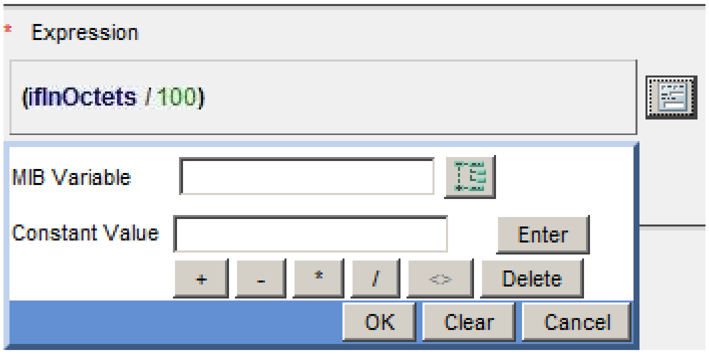
    The ifSpeed MIB Variable name should appear to the right of the divide (/) arithmetic operator you previously selected.
14. Click the rightmost * (multiply)
15. The Constant value 100 should appear to the right of the divide (/) arithmetic operator you previously selected.
16. In the Constant attribute, enter 100 and then click Enter.
17. Click **OK** to save your MIB Expression.

The following table describes each of the MIB Expression Editor options.

MIB Expression Editor Options

Attribute	Description
MIB Expression	Displays the MIB Expression as it is created. You can place the cursor in the MIB Expression field to specify where you want to add or replace an entry.
MIB Variable	You must select a MIB Variable using the MIB tree. Click the icon to access the MIB tree and navigate to the MIB variable of interest. **Note**: If you do not see a MIB that you recently loaded, wait 1 minute for NNMi to cache the new MIB information, and then open the MIB tree again. After you select a MIB Variable, NNMi displays the MIB Variable's name. If you choose a MIB Variable that has multiple instances, you MUST specify a MIB Filter Variable and MIB Filter. For example, because a node can have multiple interfaces, MIB Variables containing interface information have multiple instances, one for each interface. You are required to provide a MIB Filter value to select the interfaces you want NNMi to poll. If you do not specify a MIB Filter Variable and MIB Filter, NNMi assumes the MIB variable is non-repeating. Click here for more

The following table describes each of the MIB Expression Editor options. MIB Expression Editor Options, continued

Attribute	Description
	information. For example, if you want to always gather additional HOST-RESOURCES-MIB status information about COM (communication) port devices, you would define the following: - MIB Expression: `hrDeviceStatus` - MIB Filter Variable: `hrDeviceDescr` - MIB Filter: `COM*` See "Create a Policy" on page 449 for more information about the MIB Filter.
Constant Value	A numeric value to be used in the calculation for the MIB Expression. For example, you might want to include 100 as a constant when calculating percentages.
Enter	Includes the Constant Value in the MIB Expression.
+	Adds the results.
-	Subtracts the results.
*	Multiplies the results.
/	Divides the results.
<>	Changes the selected operator (+, -, *, and /) to the operator that appears next in sequence (from left to right) in the MIB Expression Editor. (The example below shows the operator sequence in the MIB Expression Editor.) For example, if you place your cursor at an add (+) operator in the MIB Expression, the MIB Expression Editor changes the add (+) operator to the minus (-) operator. If you place your cursor at the divide (/) operator in the MIB Expression as shown in the example below, the MIB Expression Editor changes the operator to the add (+) operator.  * Expression: (ifInOctets / 100) MIB Variable Constant Value Enter + - * / <> Delete OK Clear Cancel When using the <> (Change Operator) button, note the following:

The following table describes each of the MIB Expression Editor options. MIB Expression Editor Options, continued

Attribute	Description
	• You must select an operator in the MIB Expression before using the Change Operator (<>) button. • You can replace a MIB Variable with another MIB Variable or with a Constant. You can replace a Constant value with a MIB Variable or Constant.
Delete	Deletes the entry that is selected. If no entry is selected, NNMi deletes the last entry in the MIB Expression.
OK	Closes the MIB Expression Editor and saves your changes.
Clear	Removes any entries in the MIB Expression.
Cancel	Closes the MIB Expression Editor without saving your changes.

Override MIB OID Types

NNMi's Custom Poller determines the MIB OID Types for each MIB OID that is used in a MIB Expression or MIB Filter and lists them in the **MIB OID Types** table in the **Configuration** workspace. These MIB OID Type configurations are then used by Custom Poller, as well as the NNMi Line Graph, and the Analysis Pane Gauges feature.

Note: If you delete a MIB OID Type entry that is used by Custom Poller, MIB OID Types reappear in the MIB OID Types table the next time any of the following occurs: 1) a Custom Poller Policy is activated, 2) a Custom Polled Instance is generated, or 3) a node in the Node Group associated with a Custom Poller Policy is discovered.

If you find that the results of a MIB Expression displayed in a Line Graph or a Gauge or used by Custom Poller are not as expected, you can use the MIB OID Types configuration to override values for the following items for a MIB Object Identifier (OID):

- Primitive Type
- Whether the MIB Variable should be treated as a single instance or as multiple instances.

Tip: To view the results of Custom Poller MIB expressions, export the data to a CSV file or use Report Groups and Report Collections to export the data to NNM iSPI Performance for Metrics. See "Configure Basic Settings for a Custom Poller Collection" on page 423 and "Create a Report Group (NNM iSPI Performance for Metrics)" on page 452 for more information. You can also view the results of MIB Expressions using Line Graphs. See Monitor with Line Graphs for more information.

Reasons to override MIB OID types include correcting the Primitive Type that is provided by the MIB vendor. Or you might want Line Graph data to appear as a different Primitive Type than that displayed. For example, when viewing a Line Graph of `locIfOutputQueueDrops` and `locIfInputQueueDrops` values from the `OLD-CISCO-INTERFACES MIB`, you might notice that NNMi is graphing large numbers. To make the Line Graph more meaningful, use the MIB OID Types configuration to change the MIB OID Primitive Type from `Integer` to `Counter`.

To configure a MIB OID Type:

1. Navigate to the **Configuration** workspace.
2. Click to expand the **MIBs** folder.
3. Select **MIB OID Types**.
4. Do one of the following:
   1. To create a MIB OID Type, click the ✱ **New** icon.

      Note: You should not need to create a MIB OID Type. Custom Poller automatically generates MIB OID Types as required.
   2. To edit a MIB OID Type, double-click the row representing the configuration you want to edit.
5. Provide the required basic settings (see the MIB OID Types table).

MIB OID Type Attributes

Attribute	Description
OID (Numeric)	The numeric representation of the OID (Object Identification) value for an associated MIB variable. **Tip**: The NNMi Analysis Pane displays the textual representation of the OID for the selected MIB OID Type.
Primitive Type	Defines the base type to be used for the associated MIB variable value. Valid values include the following: • INTEGER • UNSIGNED_INTEGER • OCTET_STRING • COUNTER • COUNTER64 • GAUGE • TIME_TICKS • IP_ADDRESS
isTabular	Specifies whether the MIB variable represented by the selected OID defines multiple instances grouped in a MIB table. Enabled ☑, indicates the associated MIB variable has multiple instances. Disabled □, indicates the associated MB variable represents a single object instance.

Purchase an HP Network Node Manager i Smart Plug-in

HP Network Node Manager i Software Smart Plug-ins (iSPIs) extend NNMi capabilities.For more information about purchasing, contact your HP sales representative. For example, each NNM iSPI might do the following:

- Enhance the data that is available.
- Add new workspaces, views, and forms.
- Add tabs to existing NNMi forms.
- Change the features of the NNMi user interface.

Multiple NNM iSPIs are available, enabling you to manage your network in a way that makes sense in your organization:

- HP Network Node Manager iSPI Network Engineering Toolset Software (NNM iSPI NET)

  Tip: See the NNMi Release Notes for a description, **Help** → **Documentation Library** → **Release Notes**.
- HP Network Node Manager iSPI for IP Multicast Software
- HP Network Node Manager iSPI for MPLS Software
- HP Network Node Manager iSPI Performance for Metrics Software
- HP Network Node Manager iSPI Performance for Quality Assurance Software
- HP Network Node Manager iSPI Performance for Traffic Software
- HP Network Node Manager iSPI for IP Telephony Software

See also the documentation available for each NNM iSPI, available at: `http://h20230.www2.hp.com/selfsolve/manuals.`

Related Topics:

"Track Your NNMi Licenses" on page 1574

"Extend a Licensed Capacity" on page 1575

"Integrations with HP and Third-Party Products" on the next page

Annotate NNM iSPI Performance for Metrics Reports

You can use Custom Attributes to include additional Node or Interface information in NNM iSPI Performance for Metrics reports.

For example, you might want to add information that identifies the interface Wide Area Network circuit.

To create a Node Custom Attribute to use in NNM iSPI Performance for Metrics reports:

1. Navigate to the nodes view (for example: **Inventory** → **Nodes**).
2. Use Ctrl-Click to select each node to which you want to add a Custom Attribute.

   > **Tip:** You can also select Nodes from a map view.

3. Select **Actions** > **Custom Attributes** → **Add**.
4. In the **Name** drop-down menu, select **NPS Annotation**.

   NPS (Network Performance Server) is the database server installed with the NNM iSPI Performance for Metrics software.

5. In the **Value** attribute, enter the value you want to appear with each node included in the NNM iSPI for Metrics report.

   The maximum length is 255 characters. Alpha-numeric, spaces, and special characters (~ ! @ # $ % ^ & * ( ) _+ -) are permitted.

To create an Interface Custom Attribute to use in NNM iSPI Performance for Metrics reports:

1. Navigate to the Interfaces view (for example: **Inventory** → **Interfaces**).
2. Use Ctrl-Click to select each interface to which you want to add a Custom Attribute.

   > **Tip:** You can also select Interfaces from a map view.

3. Select **Actions** → **Custom Attributes** → **Add**.
4. In the **Name** drop-down menu, select **NPS Annotation**.
5. In the **Value** attribute, enter the value you want to appear with each interface included in the NNM iSPI for Metrics report. For example, to identify a WAN circuit, the value might be: `ATT Circuit ID 1237`.

   The maximum length is 255 characters. Alpha-numeric, spaces, and special characters (~ ! @ # $ % ^ & * ( ) _+ -) are permitted.

To remove a Custom Attribute, select **Actions** → **Custom Attributes** → **Remove**.

Integrations with HP and Third-Party Products

You can configure multiple HP and third-party software products to share data with NNMi and receive data from NNMi. You can view details about each of these integrations in separate documents. To read any of these documents, do the following:

1. Point your web browser to the following website:
   `http://h20230.www2.hp.com/selfsolve/manuals`
2. Supply your HP Passport credentials: **Sign-in to HP Passport**.
3. Select your search criteria:

- Product
- Version
- Operating System

4. Select the link to `nnmi_doc_list_9.20.pdf`
5. Click the URL to the document you want to read.

Each integration adds some or all of the following functionality (depending on the integration):

- NNMi incidents are available in the integrated product's events viewer.
- NNMi receives and monitors traps related to the integrated product.
- NNMi operators can open some of the integrated product's views from within the NNMi console. Those views are in context of the object selected in the NNMi console (for example, node or interface).
- Operators of the integrated product can open some NNMi console views from within the integrated product. Those views are in context of the object selected in the integrated product.
- Network topology (inventory) information is shared between NNMi and the integrated product.

For information about the available integrations, see **Help** → **Documentation Library** → **Release Notes** and contact your HP sales representative.

Related Topics:

"Track Your NNMi Licenses" on page 1574

"Extend a Licensed Capacity" on page 1575

"Purchase an HP Network Node Manager i Smart Plug-in" on page 1486

Integration Configuration Form

You can configure a variety of HP and third-party software products (that run independently of NNMi) to integrate with NNMi. See **Help** → **Documentation Library** → **Release Notes**, and locate the **Support Matrix** for a complete list of supported products. You can view details about each of these integrations in separate documents. To read any of these documents, do the following:

1. Point your web browser to the following website:
   `http://h20230.www2.hp.com/selfsolve/manuals`
2. Supply your HP Passport credentials: **Sign-in to HP Passport**.
3. Select your search criteria:
   - Product
   - Version
   - Operating System
4. Select the link to `nnmi_doc_list_9.20.pdf`
5. Click the URL to the document you want to read.

Each integration adds some or all of the following functionality (depending on the integration):

- NNMi incidents are available in the integrated product’s events viewer.
- NNMi receives and monitors traps related to the integrated product.
- NNMi operators can open some of the integrated product’s views from within the NNMi console. Those views are in context of the object selected in the NNMi console (for example, node or interface).
- Operators of the integrated product can open some NNMi console views from within the integrated product. Those views are in context of the object selected in the integrated product.
- Network topology (inventory) information is shared between NNMi and the integrated product.

Some of these products require that you provide information in the NNMi **Integration Module Configuration** workspace. Use the appropriate integration configuration form to provide the information required for enabling an integration between NNMi and the associated product. For the latest information about an integration and the fields on its integration configuration form, do the following:

> **Note:** HP Network Node Manager i Software Smart Plug-ins (iSPIs) do not use the Integration Module Configuration workspace. NNM iSPIs have an entirely different configuration strategy. See "Purchase an HP Network Node Manager i Smart Plug-in" on page 1486.

1. Point your web browser to the following website:
   `http://h20230.www2.hp.com/selfsolve/manuals`
2. Supply your HP Passport credentials: **Sign-in to HP Passport**.
3. Select your search criteria:
   - Product
   - Version
   - Operating System
4. Select the link to `nnmi_doc_list_9.20.pdf`
5. Click the URL to the document you want to read.

Chapter 18

Integrating NNMi Elsewhere with URLs

Use URLs to provide access to the console or certain NNMi features. For example:

- Embed views within your company Web portal.
- Launch a map from within other applications, such as from an email.
- Launch a filtered view from a browser window to quickly find the information you need.
- Run a tool without opening the console.

The URLs you write must conform to "W3C Rules for URLs" below.

For a quick-reference list of all URL choices for launching NNMi, see **Help** → **Documentation Library** → **Integrate NNMi Elsewhere with URLs**. The Integrate NNMi Elsewhere with URLs page also contains a link to a list of sample URLs that you can copy/paste as a starting point.

Prerequisite: NNMi requires authentication for access through URLs. See "Authentication Requirements for URLs Access" on the next page.

Related Topics:

"Launch the Console (showMain)" on page 1494

"Launch a View (showView)" on page 1495

"Launch a Form (showForm)" on page 1537

"Launch Menu Items (runTool)" on page 1552

"Confirm that NNMi Is Running (cmd=isRunning)" on page 1572

W3C Rules for URLs

The World Wide Web Consortium (W3C) allows only ASCII characters in URLs.

When configuring URLs, the following characters are always allowed:

- Alpha-numeric (A-Z a-z 0-9)
- - (hyphen)
- . (period)
- _ (underline)
- ~ (tilde)

Depending on the browser and the context, some characters require special formatting with Percent Encoding. A small number of possible values are shown in the quick reference table below.

You can designate the space character several ways:

- `+` (works in all browsers, recommended because it is easiest to read)
- `%20` (Percent Encoded value, works in all browsers)
- space character (works in the browsers supported by NNMi, but is not guaranteed to work in all browsers)

RFC 3986 Characters Reserved as Delimiters
(If not specifying a delimiter, use Percent-Encoding value)

Character	:	/	?	#	[	]	@	!	$
Percent Encoded	%3A	%2F	%3F	%23	%5B	%5D	%40	%21	%24
Character	&	'	(	)	*	+	,	;	=
Percent Encoded	%26	%27	%28	%29	%2A	%2B	%2C	%3B	%3D

Additional Commonly Used Characters and Their Percent Encoding

Character	space	%	<	>
Percent Encoded	%20 (or + allowed)	%25	%3C	%3E

Authentication Requirements for URLs Access

Authentication requirements for URL access to various NNMi features are the same as for signing into the NNMi console. Each user must have a preconfigured user name, password, and default **NNMi User Group**[1] mapping. For more information, see:

- "Choose a Mode for NNMi Access" on page 504
- "User Groups Provided in NNMi" on page 547
- "Configuring Sign-In to the NNMi Console" on page 581

To bypass the NNMi sign-in page, do one of the following:

- Configure your network environment with Public Key Infrastructure (PKI) user authentication.

> **Note:** If your network environment uses X.509 Certificates such as Public Key Infrastructure (PKI) user authentication, URL authentication requires a certificate (the same as accessing the main NNMi console). See , “Configuring NNMi to Support Public Key

[1] NNMi User Groups are those User Groups provided by NNMi. Users cannot access the NNMi console until their User Account is mapped to at least one of the following NNMi User Groups: NNMi Administrators, NNMi Level 2 Operators, NNMi Level 1 Operators (with less access privileges than Level 2 Operators), and NNMi Guest Users

> Infrastructure User Authentication” chapter in the *HP Network Node Manager i Software Deployment Reference* for more information, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.

- Include the following two parameters in your URL string. Any URL request that contains `j_username` and `j_password` redirects, so the actual user name and password are not visible in the Web browser:

  a. j_username

  b. j_password

  > **Caution:** There is an inherent vulnerability in passing a plain text password as a URL parameter. Consider configuring the NNMi management server to use https/SSL (secure sockets layer encryption) so that user names/passwords are encrypted between client and server. To configure the NNMi Web server to use `https` instead of `http`, see the "Working with Certificates for NNMi" chapter in the *HP Network Node Manager i Software Deployment Reference*.

  It is recommended that these parameters be used to bypass the NNMi sign-in page for only NNMi users mapped to the User Group: *NNMi Guest Users* (providing "read-only" access to a subset of console features). For example, if you have previously defined an account where both the user Name and Password are "guest", the following brings up a list of example URLs:

  ```
  http://<serverName>:<portNumber>/nnm/launch?j_username=guest&j_
  password=guest
  ```

  > **Note:** If the NNMi Web server uses the https protocol, use `https` instead of `http`. See the "Working with Certificates for NNMi" chapter in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.

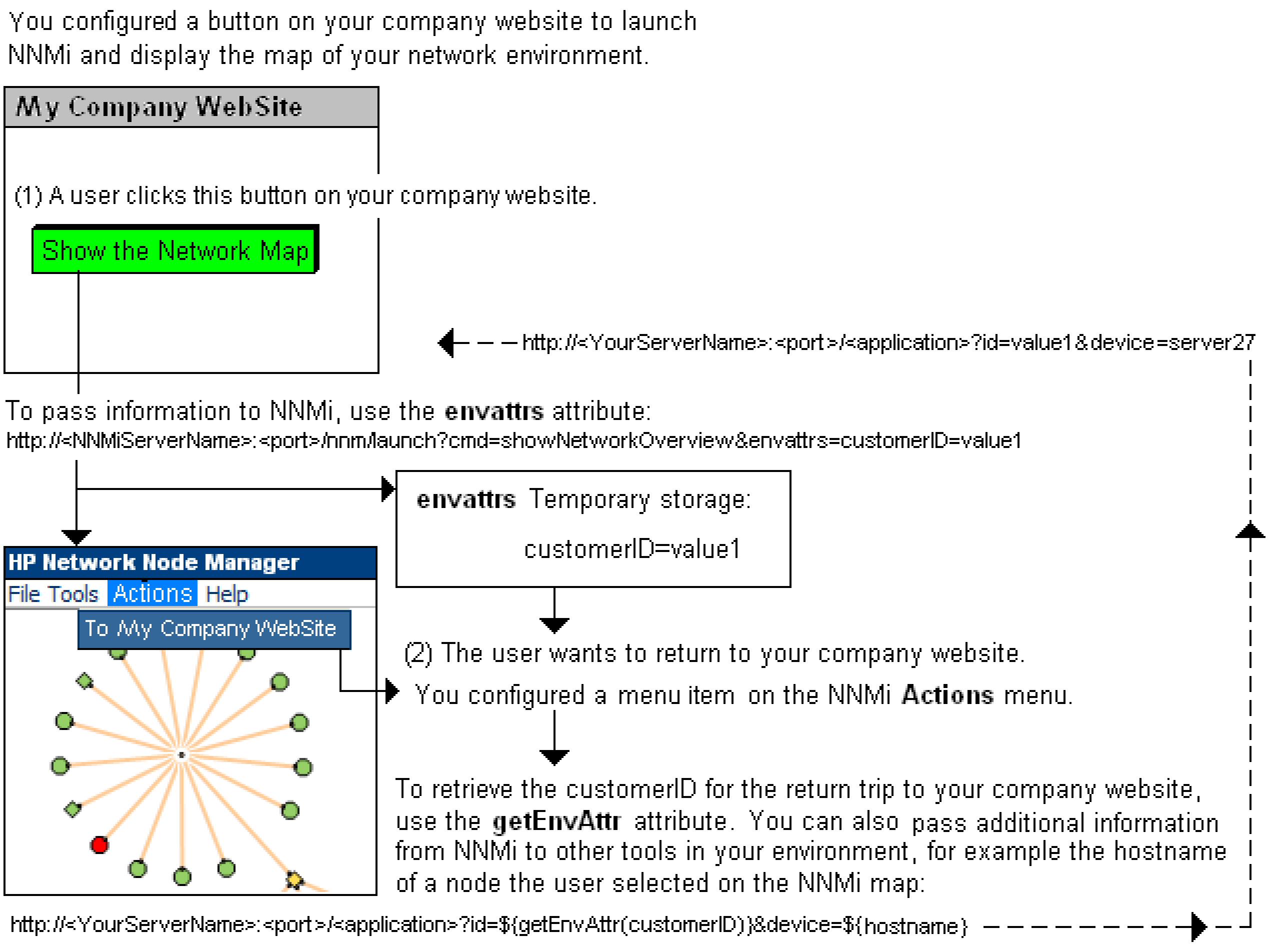
  <serverName> = the fully-qualified domain name of the NNMi management server (values allowed here are determined by the *Enable URL Redirect* setting in User Interface Configuration, see "Configuring the NNMi User Interface" on page 467)

  <portNumber> = the NNMi HTTP port number

If the user name and password are not valid, the NNMi sign-in page appears with an authentication error message.

Pass Environment Attributes

Environment Attributes (`envattrs`) are received from another application when NNMi is launched from that external application, see "Launch a View (showView)" on page 1495 or "Launch a Form (showForm)" on page 1537 for more information. These `envattrs` attributes are session-specific and not stored in the NNMi database. NNMi temporarily retains the `envattrs` name-value pairs. You can use `getEnvAttr` to retrieve a current `envattrs` value pair and pass it back to that application. Click here for an illustrated example.



You configured a button on your company website to launch NNMi and display the map of your network environment.

(1) A user clicks this button on your company website.

http://<YourServerName>:<port>/<application>?id=value1&device=server27

To pass information to NNMi, use the **envattrs** attribute:

http://<NNMiServerName>:<port>/nnm/launch?cmd=showNetworkOverview&envattrs=customerID=value1

envattrs Temporary storage:

customerID=value1

(2) The user wants to return to your company website.

You configured a menu item on the NNMi **Actions** menu.

To retrieve the customerID for the return trip to your company website, use the **getEnvAttr** attribute. You can also pass additional information from NNMi to other tools in your environment, for example the hostname of a node the user selected on the NNMi map:

http://<YourServerName>:<port>/<application>?id=${getEnvAttr(customerID)}&device=${hostname}

Note: See "Configure Launch Actions" on page 1422 for information about adding menu items to the NNMi console menus.

You can send any number of Environment Attributes (`envattrs`) when launching NNMi from another website or program. You can use `getEnvAttr` to retrieve the current `envattrs` name=value pairs and pass them back:

`${getEnvAttr(<applicationAttrName>)}`

Note: If you copy/paste this URL, remove the spaces that were added for line-ending purposes in the documentation.

```
http://<
yourServerName>:<portNumber>/<application>?<yourURLparameter1>=
${getEnvAttr(<applicationAttrName1>)}&<yourURLparameter2>=
${getEnvAttr(<applicationAttrName2>)}
```

Note: To extend the NNMi environment with additional applications, you must deploy them into a separate web-server or application-server on the same or different physical server from where the NNMi web-server or application-server is installed. See the *NNMi Developer's Toolkit* for more information.

`<serverName>` = the appropriate fully-qualified domain name

`<portNumber>` = the appropriate port number

For example, the following Full URL provides an Action within the NNMi console that returns the user to exactly the same place within your company website where the user was before launching NNMi:

```
http://<myHost>/<myApplication>?com.my.sessionId= ${getEnvAttr
(com.my.sessionId)}&com.my.objectName= $getEnvAttr(com.my.objectName)}
```

The Full URL entry could result in the following URL:

```
http://<
myHost>/<myApplication>com.my.sessionId=123&com.my.objectName=node25
```

Note: If the Environment Attribute that you request in your Action does not exist for the selected view or form, the resulting URL passes an empty string.

Launch the Console (showMain)

To launch the entire console, use the following URL:

`http://<serverName>:<portNumber>/nnm/launch?cmd=`**`showMain`**

Note: If the NNMi Web server uses the https protocol, use `https` instead of `http`. See the "Working with Certificates for NNMi" chapter in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.

`<serverName>` = the fully-qualified domain name of the NNMi management server (values allowed here are determined by the *Enable URL Redirect* setting in User Interface Configuration, see "Configuring the NNMi User Interface" on page 467)

`<portNumber>` = the NNMi HTTP port number

To launch the console and bypass log on, use the following URL:

```
http://<serverName>:<portNumber>/nnm/launch?cmd=showMain&j_
username=<accountName>&j_password=<accountPassword>
```

> **Caution:** Review in the information in "Authentication Requirements for URLs Access" on page 1491 before bypassing log on.

Launch a View (showView)

> **Tip:** This technique launches views independent of the NNMi console. When using this URL method, do not launch the view into a browser window where the NNMi console is currently running. (If you are using Mozilla Firefox, see also Configure Mozilla Firefox Timeout Interval.) To continuously display up-to-date information in your network operation center (NOC), launch an Integration URL view.

To launch a default table view that displays all instances of a specified object type, use the following URL:

```
http://<serverName>:<portNumber>/nnm/launch?cmd= showView&objtype= <x>
```

> **Note:** If the NNMi Web server uses the https protocol, use `https` instead of `http`. See the "Working with Certificates for NNMi" chapter in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.

<serverName> = the fully-qualified domain name of the NNMi management server (values allowed here are determined by the *Enable URL Redirect* setting in User Interface Configuration, see "Configuring the NNMi User Interface" on page 467)

<portNumber> = the NNMi HTTP port number

For a quick-reference list of all URL choices for launching NNMi, see **Help** → **Documentation Library** → **Integrate NNMi Elsewhere with URLs**. The Integrate NNMi Elsewhere with URLs page also contains a link to a list of sample URLs that you can copy/paste as a starting point.

> **Note**: If you are using Mozilla Firefox, click here for more information.
>
> Due to the Mozilla Firefox security implementation, launching an NNMi view with a URL has the following limitations:
>
> - The ☒ Close button does not work.
> - The File → Close menu item does not work.
>
> Use the browser buttons to close the view.

Default View for Each Object Type and Available Filters

x = objtype Value	Default View	Node Filter	Interface Filter
`Incident`	Incidents workspace, All Incidents table view	Yes	No
`Node`	Inventory workspace, Nodes table view	Yes	No
`Interface`	Inventory workspace, Interfaces table view	Yes	Yes
`IPAddress`	Inventory workspace, IP Addresses table view	Yes	Yes
`IPSubnet`	Inventory workspace, IP Subnets table view	No	No
`NodeGroup`	Inventory workspace, Node Groups table view	No	No
`InterfaceGroup`	Inventory workspace, Interface Groups table view	No	No

The following are optional filter parameters:

```
http://<serverName>:<portNumber>/nnm/launch?cmd= showView&objtype=
<x>&nodegroup= <Name>
```

```
http://<serverName>:<portNumber>/nnm/launch?cmd= showView&objtype=
<x>&ifgroup= <Name>
```

Filter by Node Group (launched Incident, Node, Interface, and IP Address views)

Attribute	Values
nodegroup	The *case-sensitive* Name attribute value of the Node Group to use as a filter for this view. **Note:** The Node Group name is translated. If your team shares NNMi within multiple locales, use nodegroupid or nodegroupuuid. If the Name value includes space characters, consider replacing the space character in your URL statement (see "W3C Rules for URLs" on page 1490).
nodegroupid	The `id` is the Unique Object Identifier (unique across the entire NNMi database). Provide the `id` of the Node Group to use as a filter for this view. This attribute value is not visible on any form within the console. To find this value, use the nnmconfigexport.ovpl command-line tool to export an XML file that includes the `id` attribute value for each object instance.
nodegroupuuid	The `uuid` is the Universally Unique Object Identifier (unique across all databases). Provide the `uuid` of the Node Group to use as a filter for this view. This attribute value is not visible on any form within the console. To find this value, use the nnmconfigexport.ovpl command-line tool to export an XML file that includes the `uuid` attribute value for each object instance.

Filter by Interface Group (launched Interface and IP Address views)

Attribute	Values
ifgroup	The *case-sensitive* Name attribute value of the Interface Group to use as a filter for this view. **Note:** The Interface Group name is translated. If your team shares NNMi within multiple locales, use ifgroupid or ifgroupuuid. If the Name value includes space characters, consider replacing the space character in your URL statement (see "W3C Rules for URLs" on page 1490).
ifgroupid	The `id` is the Unique Object Identifier (unique across the entire NNMi database). Provide the `id` of the Interface Group to use as a filter for this view. This attribute value is not visible on any form within the console. To find this value, use the nnmconfigexport.ovpl command-line tool to export an XML file that includes the `id` attribute value for each object instance.
ifgroupuuid	The `uuid` is the Universally Unique Object Identifier (unique across all databases). Provide the `uuid` of the Interface Group to use as a filter for this view. This attribute value is not visible on any form within the console. To find this value, use the nnmconfigexport.ovpl command-line tool to export an XML file that includes the `uuid` attribute value for each object instance.

The following are optional parameters:

```
http://<serverName>:<portNumber>/nnm/launch?cmd= showView&objtype=
<x>&menus= <true|false>&newWindow= <true|false>&readonly=
<true|false>&readonlynodegroupselector = <true|false>&envattrs=
<name1= value>;<name2= value>
```

Attributes for Launched Views

Attribute	Values
menus	true = Show the view menus and the ☒ Close button. If not specified, the default is true. false = Hide the view menus and the ☒ Close button to save space in the view.
newWindow	true = Display the view in a new browser window. This new window does not display the browser-specific menu bars. The browser-specific menu bars are hidden to provide the maximum amount of room for the requested view. false = Display the view within the current browser window (if not specified, the default is false).
readonly	**Caution:** The `readonly` setting overrides the

Attributes for Launched Views , continued

Attribute	Values
	`readonlynodegroupselector` setting. This means that when `readonly` is set to `true` and `readonlynodegroupselector` is set to `false`, users are able to change the Node Group filter. true = Prevents the user from doing either of the following: • Open any forms from the view • Manipulate any objects in the view (for example, delete an object) false = Enables the user to do either of the following: • Open any forms from the view • Manipulate any objects in the view (for example, delete an object)
readonlynodegroupselector	**Caution:** The `readonly` setting overrides the `readonlynodegroupselector` setting. This means that when `readonly` is set to `true` and `readonlynodegroupselector` is set to `false`, users are able to change the Node Group filter. true = Prevents the user from selecting a Node Group. **Note:** When `readonlynodegroupselector` is set to `true`, the Node Group filter selection box appears disabled. false = Enables the user to select a Node Group.
envattrs	Use Environment Attributes (`envattrs`) to pass `<`*`name=value`*`>` pairs *from an external application* to NNMi. Environment Attributes are session-specific and stored in memory (not in the NNMi database). The name-value pairs can be any arbitrary string (as required by the external program). For example, let's assume that NNMi was launched from an application that wants to preserve a session ID (123) and object attribute (node25) as follows: `http://<yourServerName/nnm?cmd= showView&objtype= Node&envattrs= com.my.sessionId= 123;com.my.objectName= node25`

Attributes for Launched Views , continued

Attribute	Values
	Note: See "Pass Environment Attributes" on page 1493 for information about how to retrieve these Environment Attributes (`envattrs`) <*name=value*> pairs from NNMi, and pass them back *to the originating external application*.

If you want to launch some other view, specify the view rather than the object type:

`http://<serverName>:<portNumber>/nnm/launch?cmd=` **`showView&view= <x>`**

For more information, see:

"Launch an Incident View" below

"Launch a Topology Maps Workspace View" on page 1504

"Launch a Monitoring Workspace View" on page 1515

"Launch a Troubleshooting Workspace View" on page 1519

"Launch an Inventory Workspace View" on page 1528

"Launch a Management Mode Workspace Views" on page 1532

"Launch a Configuration Workspace View" on page 1535

Launch an Incident View

Note: If you copy/paste this URL, remove the spaces that were added for line-ending purposes in the documentation.

`http://<serverName>:<portNumber>/nnm/launch?cmd=` **`showView&view= <x>`**

Note: If the NNMi Web server uses the https protocol, use `https` instead of `http`. See the "Working with Certificates for NNMi" chapter in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.

<serverName> = the fully-qualified domain name of the NNMi management server (values allowed here are determined by the *Enable URL Redirect* setting in User Interface Configuration, see "Configuring the NNMi User Interface" on page 467)

<portNumber> = the NNMi HTTP port number

For a quick-reference list of all URL choices for launching NNMi, see **Help → Documentation Library → Integrate NNMi Elsewhere with URLs**. The Integrate NNMi Elsewhere with URLs page also contains a link to a list of sample URLs that you can copy/paste as a starting point.

Note: If you are using Mozilla Firefox, click here for more information.

Due to the Mozilla Firefox security implementation, launching an NNMi view with a URL has the following limitations:

- The Close button does not work.
- The File → Close menu item does not work.

Use the browser buttons to close the view.

Potential Incident Workspace Views and Available Filters

View Name	*x* = View ID	Node Filter	Interface Filter
All Incidents	`allIncidentsTableView` **Tip:** To display the All Incidents view filtered by a specified node, see "Launch the All Incidents View Filtered by Node" on page 1502	Yes	No
Closed Key Incidents	`closedKeyIncidentsTableView`	Yes	No
Custom Incidents	`customIncidentTableView`	Yes	No
Custom Open Incidents	`customOpenIncidentTableView`	Yes	No
My Open Incidents	`myIncidentTableView`	Yes	No
NNM 6.x / 7.x Events	`nnm6x7xIncidentTableView`	Yes	No
Open Key Incidents	`openKeyIncidentsTableView`	Yes	No
Open Root Cause Incidents	`openRCIncidentTableView`	Yes	No
Service Impact Incidents	`serviceImpactIncidentTableView`	Yes	No
SNMP Traps	`snmpTrapsIncidentTableView`	Yes	No
Unassigned Open Key Incidents	`unassignedKeyIncidentsTableView`	Yes	No

The following are optional filter parameters:

```
http://<serverName>:<portNumber>/nnm/launch?cmd= showView&view=
<x>&nodegroup= <Name>
```

> **Note:** If you copy/paste this URL, remove the spaces that were added for line-ending purposes in the documentation.

Filter by Node Group (launched Incident, Node, Interface, and IP Address views)

Attribute	Values
nodegroup	The *case-sensitive* Name attribute value of the Node Group to use as a filter for this view. **Note:** The Node Group name is translated. If your team shares NNMi within multiple locales, use nodegroupid or nodegroupuuid. If the Name value includes space characters, consider replacing the space character in your URL statement (see "W3C Rules for URLs" on page 1490).
nodegroupid	The `id` is the Unique Object Identifier (unique across the entire NNMi database). Provide the `id` of the Node Group to use as a filter for this view. This attribute value is not visible on any form within the console. To find this value, use the nnmconfigexport.ovpl command-line tool to export an XML file that includes the `id` attribute value for each object instance.
nodegroupuuid	The `uuid` is the Universally Unique Object Identifier (unique across all databases). Provide the `uuid` of the Node Group to use as a filter for this view. This attribute value is not visible on any form within the console. To find this value, use the nnmconfigexport.ovpl command-line tool to export an XML file that includes the `uuid` attribute value for each object instance.

The following are optional parameters:

```
http://<serverName>:<portNumber>/nnm/launch?cmd= showView&view=
<x>&menus= <true|false>&newWindow= <true|false>&envattrs= <name1=
value>;<name2= value>
```

> **Note:** If you copy/paste this URL, remove the spaces that were added for line-ending purposes in the documentation.

Attributes for Launched Views

Attribute	Values
menus	true = Show the view menus and the Close button. If not specified, the default is true. false = Hide the view menus and the Close button to save space in the view.

Attributes for Launched Views , continued

Attribute	Values
newWindow	true = Display the view in a new browser window. This new window does not display the browser-specific menu bars. The browser-specific menu bars are hidden to provide the maximum amount of room for the requested view. false = Display the view within the current browser window (if not specified, the default is false).
readonly	true = Prevents the user from doing either of the following: • Open any forms from the view • Manipulate any objects in the view (for example, delete an object) false = Enables the user to do either of the following: • Open any forms from the view • Manipulate any objects in the view (for example, delete an object)
envattrs	Use Environment Attributes (`envattrs`) to pass `<name=value>` pairs *from an external application* to NNMi. Environment Attributes are session-specific and stored in memory (not in the NNMi database). The name-value pairs can be any arbitrary string (as required by the external program). For example, let's assume that NNMi was launched from an application that wants to preserve a session ID (123) and object attribute (node25) as follows: `http://<yourServerName/nnm?cmd= showView&objtype= Node&envattrs= com.my.sessionId= 123;com.my.objectName= node25` **Note:** See "Pass Environment Attributes" on page 1493 for information about how to retrieve these Environment Attributes (`envattrs`) `<name=value>` pairs from NNMi, and pass them back *to the originating external application*.

Launch the All Incidents View Filtered by Node

> **Note:** If you copy/paste this URL, remove the spaces that were added for line-ending purposes in the documentation.

```
http://<serverName>:<portNumber>/nnm/launch?cmd= showIncidents&object-
identity= <Name>
```

> **Note:** If the NNMi Web server uses the https protocol, use `https` instead of `http`. See the "Working with Certificates for NNMi" chapter in the *HP Network Node Manager i Software*

> *Deployment Reference*, which is available at:
> `http://h20230.www2.hp.com/selfsolve/manuals`.

<serverName> = the fully-qualified domain name of the NNMi management server (values allowed here are determined by the *Enable URL Redirect* setting in User Interface Configuration, see "Configuring the NNMi User Interface" on page 467)

<portNumber> = the NNMi HTTP port number

For a quick-reference list of all URL choices for launching NNMi, see **Help → Documentation Library → Integrate NNMi Elsewhere with URLs**. The Integrate NNMi Elsewhere with URLs page also contains a link to a list of sample URLs that you can copy/paste as a starting point.

> **Note**: If you are using Mozilla Firefox, click here for more information.
>
> Due to the Mozilla Firefox security implementation, launching an NNMi view with a URL has the following limitations:
>
> - The ☒ Close button does not work.
> - The File → Close menu item does not work.
>
> Use the browser buttons to close the view.

The following are optional filter parameters:

`http://`*`<serverName>`*`:`*`<portNumber>`*`/nnm/launch?cmd= showIncidents`**`&object-identity= <`*`Name`*`>`**

> **Note:** If you copy/paste this URL, remove the spaces that were added for line-ending purposes in the documentation.

Filter the All Incidents View by Node Name

Attribute	Values
Name	The *case-sensitive* Name attribute value of the Node to use as a filter for this view. **Note:** The Node Name is translated. If your team shares NNMi within multiple locales, use the `showView` command with nodegroupid or nodegroupuuid. See "Launch an Incident View" on page 1499 for more information. If the Name value includes space characters, consider replacing the space character in your URL statement (see "W3C Rules for URLs" on page 1490).

The following are optional parameters:

`http://`*`<serverName>`*`:`*`<portNumber>`*`/nnm/launch?cmd= showIncidents&object-identity= <Name>`**`&menus= <`*`true|false`*`>&newWindow= <`*`true|false`*`>&readonly= <`*`true|false`*`>&envattrs= <`*`name1= value`*`>;<`*`name2= value`*`>`**

> **Note:** If you copy/paste this URL, remove the spaces that were added for line-ending purposes in the documentation.

Attributes for Launched Views

Attribute	Values
menus	true = Show the view menus and the ☒ Close button. If not specified, the default is true. false = Hide the view menus and the ☒ Close button to save space in the view.
newWindow	true = Display the view in a new browser window. This new window does not display the browser-specific menu bars. The browser-specific menu bars are hidden to provide the maximum amount of room for the requested view. false = Display the view within the current browser window (if not specified, the default is false).
readonly	true = Prevents the user from doing either of the following: • Open any forms from the view • Manipulate any objects in the view (for example, delete an object) false = Enables the user to do either of the following: • Open any forms from the view • Manipulate any objects in the view (for example, delete an object)
envattrs	Use Environment Attributes (`envattrs`) to pass `<`*`name=value`*`>` pairs *from an external application* to NNMi. Environment Attributes are session-specific and stored in memory (not in the NNMi database). The name-value pairs can be any arbitrary string (as required by the external program). For example, let's assume that NNMi was launched from an application that wants to preserve a session ID (123) and object attribute (node25) as follows: `http://<yourServerName/nnm?cmd= showView&objtype= Node&envattrs= com.my.sessionId= 123;com.my.objectName= node25` **Note:** See "Pass Environment Attributes" on page 1493 for information about how to retrieve these Environment Attributes (`envattrs`) `<`*`name=value`*`>` pairs from NNMi, and pass them back *to the originating external application*.

Launch a Topology Maps Workspace View

The URL required for each one is unique.

> **Tip:** This technique launches views independent of the NNMi console. When using this URL method, do not launch the view into a browser window where the NNMi console is currently running. The view automatically updates every 30 seconds. This is useful if your network operation center (NOC) continuously displays a map of the most important nodes. See "Configure Maps" on page 487. (If you use the Mozilla Firefox browser and are prompted to click Continue before the map appears, see Configure Mozilla Firefox Timeout Interval.)

For a quick-reference list of all URL choices for launching NNMi, see **Help** → **Documentation Library** → **Integrate NNMi Elsewhere with URLs**. The Integrate NNMi Elsewhere with URLs page also contains a link to a list of sample URLs that you can copy/paste as a starting point.

Click here to show the example of a URL that opens the **Node Group Overview** map (cmd=showNodeGroupOverview).

```
http:/<serverName>:<portNumber>/nnm/launch?cmd= showNodeGroupOverview
```

> **Note:** If the NNMi Web server uses the https protocol, use `https` instead of `http`. See the "Working with Certificates for NNMi" chapter in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.

<serverName> = the fully-qualified domain name of the NNMi management server (values allowed here are determined by the *Enable URL Redirect* setting in User Interface Configuration, see "Configuring the NNMi User Interface" on page 467)

<portNumber> = the NNMi HTTP port number

> **Note**: If you are using Mozilla Firefox, click here for more information.
>
> Due to the Mozilla Firefox security implementation, launching an NNMi view with a URL has the following limitations:
>
> - The ☒ Close button does not work.
> - The File → Close menu item does not work.
>
> Use the browser buttons to close the view.

The following are optional parameters:

```
http://<serverName>:<portNumber>/nnm/launch?cmd=
showNodeGroupOverview&menus= <true/false>&newWindow=
<true/false>&readonly= <true|false>&envattrs= <name1= value>;<name2=
value>
```

> **Note:** If you copy/paste this URL, remove the spaces that were added for line-ending purposes in the documentation.

Attributes for Launched Views

Attribute	Values
menus	true = Show the view menus and the ☒ Close button. If not specified, the default is true. false = Hide the view menus and the ☒ Close button to save space in the view.
newWindow	true = Display the view in a new browser window. This new window does not display the browser-specific menu bars. The browser-specific menu bars are hidden to provide the maximum amount of room for the requested view. false = Display the view within the current browser window (if not specified, the default is false).
readonly	true = Prevents the user from doing either of the following: • Open any forms from the view • Manipulate any objects in the view (for example, delete an object) false = Enables the user to do either of the following: • Open any forms from the view • Manipulate any objects in the view (for example, delete an object)
envattrs	Use Environment Attributes (`envattrs`) to pass `<`*`name=value`*`>` pairs *from an external application* to NNMi. Environment Attributes are session-specific and stored in memory (not in the NNMi database). The name-value pairs can be any arbitrary string (as required by the external program). For example, let's assume that NNMi was launched from an application that wants to preserve a session ID (123) and object attribute (node25) as follows: `http://<yourServerName/nnm?cmd= showView&objtype= Node&envattrs= com.my.sessionId= 123;com.my.objectName= node25` **Note:** See "Pass Environment Attributes" on page 1493 for information about how to retrieve these Environment Attributes (`envattrs`) `<`*`name=value`*`>` pairs from NNMi, and pass them back *to the originating external application*.

Click here to show the example of a URL that opens the **Network Overview** map (cmd=showNetworkOverview).

```
http:/<serverName>:<portNumber>/nnm/launch?cmd= showNetworkOverview
```

> **Note:** If the NNMi Web server uses the https protocol, use `https` instead of `http`. See the "Working with Certificates for NNMi" chapter in the *HP Network Node Manager i Software*

> *Deployment Reference*, which is available at:
> `http://h20230.www2.hp.com/selfsolve/manuals`.

<serverName> = the fully-qualified domain name of the NNMi management server (values allowed here are determined by the *Enable URL Redirect* setting in User Interface Configuration, see "Configuring the NNMi User Interface" on page 467)

<portNumber> = the NNMi HTTP port number

For a quick-reference list of all URL choices for launching NNMi, see **Help → Documentation Library → Integrate NNMi Elsewhere with URLs**. The Integrate NNMi Elsewhere with URLs page also contains a link to a list of sample URLs that you can copy/paste as a starting point.

> **Note**: If you are using Mozilla Firefox, click here for more information.
>
> Due to the Mozilla Firefox security implementation, launching an NNMi view with a URL has the following limitations:
>
> - The ☒ Close button does not work.
> - The File → Close menu item does not work.
>
> Use the browser buttons to close the view.

The following are optional parameters:

```
http://<serverName>:<portNumber>/nnm/launch?cmd=
showNetworkOverview&menus= <true|false>&newWindow=
<true|false>&readonly= <true|false>&envattrs= <name1= value>;<name2=
value>
```

> **Note:** If you copy/paste this URL, remove the spaces that were added for line-ending purposes in the documentation.

Attributes for Launched Views

Attribute	Values
menus	true = Show the view menus and the ☒ Close button. If not specified, the default is true. false = Hide the view menus and the ☒ Close button to save space in the view.
newWindow	true = Display the view in a new browser window. This new window does not display the browser-specific menu bars. The browser-specific menu bars are hidden to provide the maximum amount of room for the requested view. false = Display the view within the current browser window (if not specified, the default is false).
readonly	true = Prevents the user from doing either of the following:

Attributes for Launched Views , continued

Attribute	Values
	- Open any forms from the view - Manipulate any objects in the view (for example, delete an object) false = Enables the user to do either of the following: - Open any forms from the view - Manipulate any objects in the view (for example, delete an object)
envattrs	Use Environment Attributes (`envattrs`) to pass `<name=value>` pairs *from an external application* to NNMi. Environment Attributes are session-specific and stored in memory (not in the NNMi database). The name-value pairs can be any arbitrary string (as required by the external program). For example, let's assume that NNMi was launched from an application that wants to preserve a session ID (123) and object attribute (node25) as follows: `http://<yourServerName/nnm?cmd= showView&objtype= Node&envattrs= com.my.sessionId= 123;com.my.objectName= node25` **Note:** See "Pass Environment Attributes" on page 1493 for information about how to retrieve these Environment Attributes (`envattrs`) `<name=value>` pairs from NNMi, and pass them back *to the originating external application*.

Click here to show the example of a URL that opens the **Networking Infrastructure Devices** node group map (cmd=showView).

See quick reference "W3C Rules for URLs" on page 1490.

```
http:/<serverName>:<portNumber>/nnm/launch?cmd= showView&objtype=
Node&nodegroup= Networking+Infrastructure+Devices
```

Note: If the NNMi Web server uses the https protocol, use `https` instead of `http`. See the "Working with Certificates for NNMi" chapter in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.

<serverName> = the fully-qualified domain name of the NNMi management server (values allowed here are determined by the *Enable URL Redirect* setting in User Interface Configuration, see "Configuring the NNMi User Interface" on page 467)

<portNumber> = the NNMi HTTP port number

Note: If you are using Mozilla Firefox, click here for more information.

Due to the Mozilla Firefox security implementation, launching an NNMi view with a URL has the following limitations:

- The ☒ Close button does not work.
- The File → Close menu item does not work.

Use the browser buttons to close the view.

The following are optional parameters:

```
http://<serverName>:<portNumber>/nnm/launch?cmd= showView&objtype=
Node&nodegroup= Networking+Infrastructure+Devices&menus=
<true|false>&newWindow= <true|false>&readonly=
<true|false>&readonlynodegroupselector= <true|false>&envattrs= <name1=
value>;<name2= value>
```

Note: If you copy/paste this URL, remove the spaces that were added for line-ending purposes in the documentation.

Attributes for Launched Views

Attribute	Values
menus	true = Show the view menus and the ☒ Close button. If not specified, the default is true. false = Hide the view menus and the ☒ Close button to save space in the view.
newWindow	true = Display the view in a new browser window. This new window does not display the browser-specific menu bars. The browser-specific menu bars are hidden to provide the maximum amount of room for the requested view. false = Display the view within the current browser window (if not specified, the default is false).
readonly	**Caution:** The `readonly` setting overrides the `readonlynodegroupselector` setting. This means that when `readonly` is set to `true` and `readonlynodegroupselector` is set to `false`, users are able to change the Node Group filter. true = Prevents the user from doing either of the following: • Open any forms from the view • Manipulate any objects in the view (for example, delete an object)

Attributes for Launched Views , continued

Attribute	Values
	false = Enables the user to do either of the following: • Open any forms from the view • Manipulate any objects in the view (for example, delete an object)
readonlynodegroupselector	**Caution:** The `readonly` setting overrides the `readonlynodegroupselector` setting. This means that when `readonly` is set to `true` and `readonlynodegroupselector` is set to `false`, users are able to change the Node Group filter. true = Prevents the user from selecting a Node Group. **Note:** When `readonlynodegroupselector` is set to `true`, the Node Group filter selection box appears disabled. false = Enables the user to select a Node Group.
envattrs	Use Environment Attributes (`envattrs`) to pass `<`*`name=value`*`>` pairs *from an external application* to NNMi. Environment Attributes are session-specific and stored in memory (not in the NNMi database). The name-value pairs can be any arbitrary string (as required by the external program). For example, let's assume that NNMi was launched from an application that wants to preserve a session ID (123) and object attribute (node25) as follows: `http://<yourServerName/nnm?cmd= showView&objtype= Node&envattrs= com.my.sessionId= 123;com.my.objectName= node25` **Note:** See "Pass Environment Attributes" on page 1493 for information about how to retrieve these Environment Attributes (`envattrs`) `<`*`name=value`*`>` pairs from NNMi, and pass them back *to the originating external application*.

Click here to show the example of a URL that opens the **Routers** node group map (cmd=showNodeGroup&name=Routers).

```
http:/<serverName>:<portNumber>/nnm/launch?cmd= showNodeGroup&name=
Routers
```

> **Note:** If the NNMi Web server uses the https protocol, use `https` instead of `http`. See the "Working with Certificates for NNMi" chapter in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.

<serverName> = the fully-qualified domain name of the NNMi management server (values allowed here are determined by the *Enable URL Redirect* setting in User Interface Configuration, see "Configuring the NNMi User Interface" on page 467)

<portNumber> = the NNMi HTTP port number

> **Note**: If you are using Mozilla Firefox, click here for more information.
>
> Due to the Mozilla Firefox security implementation, launching an NNMi view with a URL has the following limitations:
>
> - The Close button does not work.
> - The File → Close menu item does not work.
>
> Use the browser buttons to close the view.

The following are optional parameters:

```
http://<serverName>:<portNumber>/nnm/launch?cmd= showNodeGroup&name=
Routers&menus= <true|false>&newWindow= <true|false>&readonly=
<true|false>&readonlynodegroupselector= <true|false>&envattrs= <name1=
value>;<name2= value>
```

> **Note:** If you copy/paste this URL, remove the spaces that were added for line-ending purposes in the documentation.

Attributes for Launched Views

Attribute	Values
menus	true = Show the view menus and the Close button. If not specified, the default is true. false = Hide the view menus and the Close button to save space in the view.
newWindow	true = Display the view in a new browser window. This new window does not display the browser-specific menu bars. The browser-specific menu bars are hidden to provide the maximum amount of room for the requested view. false = Display the view within the current browser window (if not specified, the default is false).

Attributes for Launched Views , continued

Attribute	Values
readonly	**Caution:** The `readonly` setting overrides the `readonlynodegroupselector` setting. This means that when `readonly` is set to `true` and `readonlynodegroupselector` is set to `false`, users are able to change the Node Group filter. true = Prevents the user from doing either of the following: • Open any forms from the view • Manipulate any objects in the view (for example, delete an object) false = Enables the user to do either of the following: • Open any forms from the view • Manipulate any objects in the view (for example, delete an object)
readonlynodegroupselector	**Caution:** The `readonly` setting overrides the `readonlynodegroupselector` setting. This means that when `readonly` is set to `true` and `readonlynodegroupselector` is set to `false`, users are able to change the Node Group filter. true = Prevents the user from selecting a Node Group. **Note:** When `readonlynodegroupselector` is set to `true`, the Node Group filter selection box appears disabled. false = Enables the user to select a Node Group.
envattrs	Use Environment Attributes (`envattrs`) to pass `<`*`name=value`*`>` pairs *from an external application* to NNMi. Environment Attributes are session-specific and stored in memory (not in the NNMi database). The name-value pairs can be any arbitrary string (as required by the external program). For example, let’s assume that NNMi was launched from an application that wants to preserve a session ID (123) and object attribute (node25) as follows: `http://<yourServerName/nnm?cmd=`

Attributes for Launched Views , continued

Attribute	Values
	`showView&objtype= Node&envattrs= com.my.sessionId= 123;com.my.objectName= node25` **Note:** See "Pass Environment Attributes" on page 1493 for information about how to retrieve these Environment Attributes (`envattrs`) <*name=value*> pairs from NNMi, and pass them back *to the originating external application*.

Click here to show the example of a URL that opens the **Switches** node group map (cmd=showNodeGroup&name=Switches).

http:/<*serverName*>:<*portNumber*>/nnm/launch?**cmd= showNodeGroup&name= Switches**

> **Note:** If the NNMi Web server uses the https protocol, use `https` instead of `http`. See the "Working with Certificates for NNMi" chapter in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.

<*serverName*> = the fully-qualified domain name of the NNMi management server (values allowed here are determined by the *Enable URL Redirect* setting in User Interface Configuration, see "Configuring the NNMi User Interface" on page 467)

<*portNumber*> = the NNMi HTTP port number

> **Note**: If you are using Mozilla Firefox, click here for more information.
>
> Due to the Mozilla Firefox security implementation, launching an NNMi view with a URL has the following limitations:
>
> - The ☒ Close button does not work.
> - The File → Close menu item does not work.
>
> Use the browser buttons to close the view.

The following are optional parameters:

http://<*serverName*>:<*portNumber*>/nnm/launch?cmd= showNodeGroup&name= Switches**&menus= <*true*|*false*>&newWindow= <*true*|*false*>&readonly= <*true*|*false*>&readonlynodegroupselector= <*true*|*false*>&envattrs= <*name1= value*>;<*name2= value*>**

> **Note:** If you copy/paste this URL, remove the spaces that were added for line-ending purposes in the documentation.

Attributes for Launched Views

Attribute	Values
menus	true = Show the view menus and the Close button. If not specified, the default is true. false = Hide the view menus and the Close button to save space in the view.
newWindow	true = Display the view in a new browser window. This new window does not display the browser-specific menu bars. The browser-specific menu bars are hidden to provide the maximum amount of room for the requested view. false = Display the view within the current browser window (if not specified, the default is false).
readonly	**Caution:** The `readonly` setting overrides the `readonlynodegroupselector` setting. This means that when `readonly` is set to `true` and `readonlynodegroupselector` is set to `false`, users are able to change the Node Group filter. true = Prevents the user from doing either of the following: • Open any forms from the view • Manipulate any objects in the view (for example, delete an object) false = Enables the user to do either of the following: • Open any forms from the view • Manipulate any objects in the view (for example, delete an object)
readonlynodegroupselector	**Caution:** The `readonly` setting overrides the `readonlynodegroupselector` setting. This means that when `readonly` is set to `true` and `readonlynodegroupselector` is set to `false`, users are able to change the Node Group filter. true = Prevents the user from selecting a Node Group. **Note:** When `readonlynodegroupselector` is set to `true`, the Node Group filter selection box appears disabled.

Attributes for Launched Views , continued

Attribute	Values
	false = Enables the user to select a Node Group.
envattrs	Use Environment Attributes (`envattrs`) to pass *<name=value>* pairs *from an external application* to NNMi. Environment Attributes are session-specific and stored in memory (not in the NNMi database). The name-value pairs can be any arbitrary string (as required by the external program). For example, let's assume that NNMi was launched from an application that wants to preserve a session ID (123) and object attribute (node25) as follows: `http://<yourServerName/nnm?cmd= showView&objtype= Node&envattrs= com.my.sessionId= 123;com.my.objectName= node25` **Note:** See "Pass Environment Attributes" on page 1493 for information about how to retrieve these Environment Attributes (`envattrs`) *<name=value>* pairs from NNMi, and pass them back *to the originating external application*.

Launch a Monitoring Workspace View

Note: If you copy/paste this URL, remove the spaces that were added for line-ending purposes in the documentation.

```
http://<serverName>:<portNumber>/nnm/launch?cmd= showView&view= <x>
```

Note: If the NNMi Web server uses the https protocol, use `https` instead of `http`. See the "Working with Certificates for NNMi" chapter in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.

<serverName> = the fully-qualified domain name of the NNMi management server (values allowed here are determined by the *Enable URL Redirect* setting in User Interface Configuration, see "Configuring the NNMi User Interface" on page 467)

<portNumber> = the NNMi HTTP port number

For a quick-reference list of all URL choices for launching NNMi, see **Help** → **Documentation Library** → **Integrate NNMi Elsewhere with URLs**. The Integrate NNMi Elsewhere with URLs page also contains a link to a list of sample URLs that you can copy/paste as a starting point.

Note: If you are using Mozilla Firefox, click here for more information.

Due to the Mozilla Firefox security implementation, launching an NNMi view with a URL has the following limitations:

- The ☒ Close button does not work.
- The File → Close menu item does not work.

Use the browser buttons to close the view.

Monitoring Workspace Views and Available Filters

View Name	*x* = View ID	Node Filter	Interface Filter
Non-Normal Node Components	`nonNormalNodeComponentTableView`	No	No
Non-Normal Cards	`nonNormalCardTableView`	No	No
+Non-Normal Interfaces	`nonNormalInterfaceTableView`	Yes	Yes
+Non-Normal Nodes	`nonNormalNodeTableView`	Yes	No
+Not Responding Addresses	`notRespondingIPAddressTableView`	Yes	Yes
Interface Performance	`interfacePerformanceTableView`	Yes	Yes
Card Redundancy Groups	`cardRedundancyGroupsTableView`	No	No
Router Redundancy Groups	`routerRedundancyGroupsStatusTableView`	No	No
Node Groups	`nodeGroupsStatusTableView`	No	No
Custom Node Collections	`customPollerNodeCollectionsTableView`	No	No
Custom Polled Instances	`customPollerPolledInstancesTableView`	No	No

The following are optional filter parameters:

```
http://<serverName>:<portNumber>/nnm/launch?cmd= showView&view=
<x>&nodegroup= <Name>
```

```
http://<serverName>:<portNumber>/nnm/launch?cmd= showView&view=
<x>&ifgroup= <Name>
```

Note: If you copy/paste this URL, remove the spaces that were added for line-ending purposes in the documentation.

Filter by Node Group (launched Incident, Node, Interface, and IP Address views)

Attribute	Values
nodegroup	The *case-sensitive* Name attribute value of the Node Group to use as a filter for this view. **Note:** The Node Group name is translated. If your team shares NNMi within multiple locales, use nodegroupid or nodegroupuuid. If the Name value includes space characters, consider replacing the space character in your URL statement (see "W3C Rules for URLs" on page 1490).
nodegroupid	The `id` is the Unique Object Identifier (unique across the entire NNMi database). Provide the `id` of the Node Group to use as a filter for this view. This attribute value is not visible on any form within the console. To find this value, use the nnmconfigexport.ovpl command-line tool to export an XML file that includes the `id` attribute value for each object instance.
nodegroupuuid	The `uuid` is the Universally Unique Object Identifier (unique across all databases). Provide the `uuid` of the Node Group to use as a filter for this view. This attribute value is not visible on any form within the console. To find this value, use the nnmconfigexport.ovpl command-line tool to export an XML file that includes the `uuid` attribute value for each object instance.

Filter by Interface Group (launched Interface and IP Address views)

Attribute	Values
ifgroup	The *case-sensitive* Name attribute value of the Interface Group to use as a filter for this view. **Note:** The Interface Group name is translated. If your team shares NNMi within multiple locales, use ifgroupid or ifgroupuuid. If the Name value includes space characters, consider replacing the space character in your URL statement (see "W3C Rules for URLs" on page 1490).
ifgroupid	The `id` is the Unique Object Identifier (unique across the entire NNMi database). Provide the `id` of the Interface Group to use as a filter for this view. This attribute value is not visible on any form within the console. To find this value, use the nnmconfigexport.ovpl command-line tool to export an XML file that includes the `id` attribute value for each object instance.
ifgroupuuid	The `uuid` is the Universally Unique Object Identifier (unique across all databases). Provide the `uuid` of the Interface Group to use as a filter for this view. This attribute value is not visible on any form within the console. To find this value, use the nnmconfigexport.ovpl command-line tool to export an XML file that includes the `uuid` attribute value for each object instance.

The following are optional parameters:

```
http://<serverName>:<portNumber>/nnm/launch?cmd= showView&view=
<x>&menus= <true|false>&newWindow= <true|false>&envattrs= <name1=
value>;<name2= value>
```

> **Note:** If you copy/paste this URL, remove the spaces that were added for line-ending purposes in the documentation.

Attributes for Launched Views

Attribute	Values
menus	true = Show the view menus and the ☒ Close button. If not specified, the default is true. false = Hide the view menus and the ☒ Close button to save space in the view.
newWindow	true = Display the view in a new browser window. This new window does not display the browser-specific menu bars. The browser-specific menu bars are hidden to provide the maximum amount of room for the requested view. false = Display the view within the current browser window (if not specified, the default is false).
readonly	true = Prevents the user from doing either of the following: • Open any forms from the view • Manipulate any objects in the view (for example, delete an object) false = Enables the user to do either of the following: • Open any forms from the view • Manipulate any objects in the view (for example, delete an object)
envattrs	Use Environment Attributes (`envattrs`) to pass `<name=value>` pairs *from an external application* to NNMi. Environment Attributes are session-specific and stored in memory (not in the NNMi database). The name-value pairs can be any arbitrary string (as required by the external program). For example, let's assume that NNMi was launched from an application that wants to preserve a session ID (123) and object attribute (node25) as follows: `http://<yourServerName/nnm?cmd= showView&objtype= Node&envattrs= com.my.sessionId= 123;com.my.objectName= node25` **Note:** See "Pass Environment Attributes" on page 1493 for information about how to retrieve these Environment Attributes (`envattrs`) `<name=value>` pairs from NNMi, and pass them back *to the originating external application*.

Launch a Troubleshooting Workspace View

There are four types of views in the Troubleshooting workspace. The URL syntax required for each one is unique.

> **Tip:** This technique launches views independent of the NNMi console. When using this URL method, do not launch the view into a browser window where the NNMi console is currently running. The view automatically updates every 30 seconds. This is useful if your network operation center (NOC) continuously displays a map of the most important nodes. See "Configure Maps" on page 487. (If you use the Mozilla Firefox browser and are prompted to click Continue before the map appears, see Configure Mozilla Firefox Timeout Interval.)

Click here to show examples of URLs that open a **Layer 2 Neighbor View** (cmd=showLayer2Neighbors).

> **Note:** If you copy/paste this URL, remove the spaces that were added for line-ending purposes in the documentation.

```
http://<serverName>:<portNumber>/nnm/launch?cmd= showLayer2Neighbors
```

```
http://<serverName>:<portNumber>/nnm/launch?cmd=
showLayer2Neighbors&nodename= <x>&hops= <#>
```

> **Note:** If the NNMi Web server uses the https protocol, use `https` instead of `http`. See the "Working with Certificates for NNMi" chapter in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.

<serverName> = the fully-qualified domain name of the NNMi management server (values allowed here are determined by the *Enable URL Redirect* setting in User Interface Configuration, see "Configuring the NNMi User Interface" on page 467)

<portNumber> = the NNMi HTTP port number

For a quick-reference list of all URL choices for launching NNMi, see **Help → Documentation Library → Integrate NNMi Elsewhere with URLs**. The Integrate NNMi Elsewhere with URLs page also contains a link to a list of sample URLs that you can copy/paste as a starting point.

> **Note**: If you are using Mozilla Firefox, click here for more information.
>
> Due to the Mozilla Firefox security implementation, launching an NNMi view with a URL has the following limitations:
>
> - The Close button does not work.
> - The File → Close menu item does not work.
>
> Use the browser buttons to close the view.

Layer 2 Neighbor View Attributes

Attribute	Value
nodename	The source node's DNS hostname (full or short) or IP address. If you use this attribute, NNMi tries to match the string you provide by following this procedure: • Check the value of the Hostname (*case-sensitive*) on the Node form. • Check the values in the Address column of the table on the Node form, Addresses tab, • Check the value of the System Name field on the in the Node form, General tab. • Check the value in the Name field on the Node form.
hops	1 - 9

The following are optional parameters:

```
http://<serverName>:<portNumber>/nnm/launch?cmd=
showLayer2Neighbors&menus= <true|false>&newWindow=
<true|false>&readonly= <true|false>&envattrs= <name1= value>;<name2=
value>
```

Note: If you copy/paste this URL, remove the spaces that were added for line-ending purposes in the documentation.

Attributes for Launched Views

Attribute	Values
menus	true = Show the view menus and the Close button. If not specified, the default is true. false = Hide the view menus and the Close button to save space in the view.
newWindow	true = Display the view in a new browser window. This new window does not display the browser-specific menu bars. The browser-specific menu bars are hidden to provide the maximum amount of room for the requested view. false = Display the view within the current browser window (if not specified, the default is false).
readonly	true = Prevents the user from doing either of the following: • Open any forms from the view • Manipulate any objects in the view (for example, delete an object) false = Enables the user to do either of the following:

Attributes for Launched Views , continued

Attribute	Values
	• Open any forms from the view • Manipulate any objects in the view (for example, delete an object)
envattrs	Use Environment Attributes (`envattrs`) to pass `<`*`name=value`*`>` pairs *from an external application* to NNMi. Environment Attributes are session-specific and stored in memory (not in the NNMi database). The name-value pairs can be any arbitrary string (as required by the external program). For example, let's assume that NNMi was launched from an application that wants to preserve a session ID (123) and object attribute (node25) as follows: `http://<yourServerName/nnm?cmd= showView&objtype= Node&envattrs= com.my.sessionId= 123;com.my.objectName= node25` **Note:** See "Pass Environment Attributes" on page 1493 for information about how to retrieve these Environment Attributes (`envattrs`) `<`*`name=value`*`>` pairs from NNMi, and pass them back *to the originating external application*.

Click here to show examples of URLs that open a **Layer 3 Neighbor View** (cmd=showLayer3Neighbors).

> **Note:** If the NNMi Web server uses the https protocol, use `https` instead of `http`. See the "Working with Certificates for NNMi" chapter in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.

```
http://<serverName>:<portNumber>/nnm/launch?cmd= showLayer3Neighbors
```

```
http://<serverName>:<portNumber>/nnm/launch?cmd=
showLayer3Neighbors&nodename= <x>&hops= <#>
```

> **Note:** If you copy/paste this URL, remove the spaces that were added for line-ending purposes in the documentation.

> **Note**: If you are using Mozilla Firefox, click here for more information.
>
> Due to the Mozilla Firefox security implementation, launching an NNMi view with a URL has the following limitations:
>
> - The ☒ Close button does not work.
> - The File → Close menu item does not work.
>
> Use the browser buttons to close the view.

Layer 3 Neighbor View Attributes

Attribute	Value
nodename	The source node's DNS hostname (full or short) or IP address. If you use this attribute, NNMi tries to match the string you provide by following this procedure: • Check the value of the Hostname (*case-sensitive*) on the Node form. • Check the values in the Address column of the table on the Node form, Addresses tab, • Check the value of the System Name field on the in the Node form, General tab. • Check the value in the Name field on the Node form.
hops	1 - 9
menus	true = Show the menus and window toolbar in the form. If not specified, the default is true. false = Hide the menus and window toolbar in the view.

The following are optional parameters:

```
http://<serverName>:<portNumber>/nnm/launch?cmd=
showLayer3Neighbors&menus= <true|false>&newWindow=
<true|false>&readonly= <true|false>&envattrs= <name1= value>;<name2=
value>
```

Note: If you copy/paste this URL, remove the spaces that were added for line-ending purposes in the documentation.

Attributes for Launched Views

Attribute	Values
menus	true = Show the view menus and the ☒ Close button. If not specified, the default is true. false = Hide the view menus and the ☒ Close button to save space in the view.
newWindow	true = Display the view in a new browser window. This new window does not display the browser-specific menu bars. The browser-specific menu bars are hidden to provide the maximum amount of room for the requested view. false = Display the view within the current browser window (if not specified, the default is false).
readonly	true = Prevents the user from doing either of the following: • Open any forms from the view

Attributes for Launched Views , continued

Attribute	Values
	• Manipulate any objects in the view (for example, delete an object) false = Enables the user to do either of the following: • Open any forms from the view • Manipulate any objects in the view (for example, delete an object)
envattrs	Use Environment Attributes (`envattrs`) to pass `<`*`name=value`*`>` pairs *from an external application* to NNMi. Environment Attributes are session-specific and stored in memory (not in the NNMi database). The name-value pairs can be any arbitrary string (as required by the external program). For example, let's assume that NNMi was launched from an application that wants to preserve a session ID (123) and object attribute (node25) as follows: `http://<yourServerName/nnm?cmd= showView&objtype= Node&envattrs= com.my.sessionId= 123;com.my.objectName= node25` **Note:** See "Pass Environment Attributes" on page 1493 for information about how to retrieve these Environment Attributes (`envattrs`) `<`*`name=value`*`>` pairs from NNMi, and pass them back *to the originating external application*.

Click here to show examples of URLs that open a **Path View** (cmd=showPath).

> **Note:** If the NNMi Web server uses the https protocol, use `https` instead of `http`. See the "Working with Certificates for NNMi" chapter in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.

`http://<`*`serverName`*`>:<`*`portNumber`*`>/nnm/launch?cmd=` **`showPath`**

`http://<`*`serverName`*`>:<`*`portNumber`*`>/nnm/launch?cmd=` **`showPath&src= <`*`x`*`>&dest= <`*`y`*`>`**

> **Note:** If you copy/paste this URL, remove the spaces that were added for line-ending purposes in the documentation.

> **Note**: If you are using Mozilla Firefox, click here for more information.
>
> Due to the Mozilla Firefox security implementation, launching an NNMi view with a URL has the following limitations:

- The Close button does not work.
- The File → Close menu item does not work.

Use the browser buttons to close the view.

Note: *NNMi Advanced*. Path View works only with IPv4 addresses. The NNMi Advanced IPv6 address values are not valid choices for Path View. Any devices in your network that are configured with IPv6 addresses cannot be displayed on Path View maps.

Path View Attributes

Attribute	Value
src	The source node's DNS hostname (full or short) or IP address. NNMi tries to match the string you provide by following this procedure: • Check the value of the Hostname (*case-sensitive*) on the Node form. • Check the values in the Address column of the table on the Node form, Addresses tab, • Check the value of the System Name field on the in the Node form, General tab. • Check the value in the Name field on the Node form.
dest	The destination node's DNS hostname (full or short) or IP address. NNMi tries to match the string you provide by following this procedure: • Check the value of the Hostname (*case-sensitive*) on the Node form. • Check the values in the Address column of the table on the Node form, Addresses tab, • Check the value of the System Name field on the in the Node form, General tab. • Check the value in the Name field on the Node form.

The following are optional parameters:

```
http://<serverName>:<portNumber>/nnm/launch?cmd= showPath&menus=
<true|false>&newWindow= <true|false>&readonly= <true|false>&envattrs=
<name1= value>;<name2= value>
```

Note: If you copy/paste this URL, remove the spaces that were added for line-ending purposes in the documentation.

Attributes for Launched Views

Attribute	Values
menus	true = Show the view menus and the Close button. If not specified, the default

Attributes for Launched Views , continued

Attribute	Values
	is true. false = Hide the view menus and the Close button to save space in the view.
newWindow	true = Display the view in a new browser window. This new window does not display the browser-specific menu bars. The browser-specific menu bars are hidden to provide the maximum amount of room for the requested view. false = Display the view within the current browser window (if not specified, the default is false).
readonly	true = Prevents the user from doing either of the following: • Open any forms from the view • Manipulate any objects in the view (for example, delete an object) false = Enables the user to do either of the following: • Open any forms from the view • Manipulate any objects in the view (for example, delete an object)
envattrs	Use Environment Attributes (`envattrs`) to pass `<`*`name=value`*`>` pairs *from an external application* to NNMi. Environment Attributes are session-specific and stored in memory (not in the NNMi database). The name-value pairs can be any arbitrary string (as required by the external program). For example, let's assume that NNMi was launched from an application that wants to preserve a session ID (123) and object attribute (node25) as follows: `http://<yourServerName/nnm?cmd= showView&objtype= Node&envattrs= com.my.sessionId= 123;com.my.objectName= node25` **Note:** See "Pass Environment Attributes" on page 1493 for information about how to retrieve these Environment Attributes (`envattrs`) `<`*`name=value`*`>` pairs from NNMi, and pass them back *to the originating external application*.

Click here to show examples of URLs that open a **Node Group Map View** (cmd=showNodeGroup).

Note: If the NNMi Web server uses the https protocol, use `https` instead of `http`. See the "Working with Certificates for NNMi" chapter in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.

http://*<serverName>*:*<portNumber>*/nnm/launch?cmd= **showNodeGroup&name= *<x>***

> **Note**: If you are using Mozilla Firefox, click here for more information.
>
> Due to the Mozilla Firefox security implementation, launching an NNMi view with a URL has the following limitations:
>
> - The ☒ Close button does not work.
> - The File → Close menu item does not work.
>
> Use the browser buttons to close the view.

Node Group Map View Attributes

Attribute	Value
name	The *case-sensitive* Name attribute value from the Node Group form. **Note:** The Node Group name is translated. If your team shares NNMi within multiple locales, use nodegroupid or nodegroupuuid. If the Name value includes space characters, consider replacing the space character in your URL statement (see "W3C Rules for URLs" on page 1490).
objid	The `id` is the Unique Object Identifier (unique per object type in the NNMi database). Provide the `id` of the Node Group to use as a filter for this view. NNMi displays the `id` attribute value on the object form's Registration tab.
objuuid	The `uuid` is the Universally Unique Object Identifier (unique across all databases). Provide the `uuid` of the Node Group to use as a filter for this view. NNMi displays the `uuid` attribute value on the object form's Registration tab.

The following are optional parameters:

http://*<serverName>*:*<portNumber>*/nnm/launch?cmd= showNodeGroup&name= *<x>***&menus= <*true*|*false*>&newWindow= <*true*|*false*>&readonly= <*true*|*false*>&readonlynodegroupselector= <*true*|*false*>&envattrs= <*name1= value*>;<*name2= value*>**

> **Note:** If you copy/paste this URL, remove the spaces that were added for line-ending purposes in the documentation.

Attributes for Launched Views

Attribute	Values
menus	true = Show the view menus and the ☒ Close button. If not specified, the default is true.

Attributes for Launched Views , continued

Attribute	Values
	false = Hide the view menus and the ☒ Close button to save space in the view.
newWindow	true = Display the view in a new browser window. This new window does not display the browser-specific menu bars. The browser-specific menu bars are hidden to provide the maximum amount of room for the requested view. false = Display the view within the current browser window (if not specified, the default is false).
readonly	**Caution:** The `readonly` setting overrides the `readonlynodegroupselector` setting. This means that when `readonly` is set to `true` and `readonlynodegroupselector` is set to `false`, users are able to change the Node Group filter. true = Prevents the user from doing either of the following: • Open any forms from the view • Manipulate any objects in the view (for example, delete an object) false = Enables the user to do either of the following: • Open any forms from the view • Manipulate any objects in the view (for example, delete an object)
readonlynodegroupselector	**Caution:** The `readonly` setting overrides the `readonlynodegroupselector` setting. This means that when `readonly` is set to `true` and `readonlynodegroupselector` is set to `false`, users are able to change the Node Group filter. true = Prevents the user from selecting a Node Group. **Note:** When `readonlynodegroupselector` is set to `true`, the Node Group filter selection box appears disabled. false = Enables the user to select a Node Group.

Attributes for Launched Views , continued

Attribute	Values
envattrs	Use Environment Attributes (`envattrs`) to pass <*name=value*> pairs *from an external application* to NNMi. Environment Attributes are session-specific and stored in memory (not in the NNMi database). The name-value pairs can be any arbitrary string (as required by the external program). For example, let's assume that NNMi was launched from an application that wants to preserve a session ID (123) and object attribute (node25) as follows: `http://<yourServerName/nnm?cmd= showView&objtype= Node&envattrs= com.my.sessionId= 123;com.my.objectName= node25` **Note:** See "Pass Environment Attributes" on page 1493 for information about how to retrieve these Environment Attributes (`envattrs`) <*name=value*> pairs from NNMi, and pass them back *to the originating external application*.

Launch an Inventory Workspace View

Note: If you copy/paste this URL, remove the spaces that were added for line-ending purposes in the documentation.

`http://<serverName>:<portNumber>/nnm/launch?cmd=` **`showView&view= <x>`**

Note: If the NNMi Web server uses the https protocol, use `https` instead of `http`. See the "Working with Certificates for NNMi" chapter in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.

<*serverName*> = the fully-qualified domain name of the NNMi management server (values allowed here are determined by the *Enable URL Redirect* setting in User Interface Configuration, see "Configuring the NNMi User Interface" on page 467)

<*portNumber*> = the NNMi HTTP port number

For a quick-reference list of all URL choices for launching NNMi, see **Help → Documentation Library → Integrate NNMi Elsewhere with URLs**. The Integrate NNMi Elsewhere with URLs page also contains a link to a list of sample URLs that you can copy/paste as a starting point.

Note: If you are using Mozilla Firefox, click here for more information.

Due to the Mozilla Firefox security implementation, launching an NNMi view with a URL has the following limitations:

- The Close button does not work.
- The File → Close menu item does not work.

Use the browser buttons to close the view.

Inventory Workspace Views and Available Filters

View Name	*x* = View ID	Node Filter	Interface Filter
Nodes	`allNodesTableView`	Yes	No
Interfaces	`allInterfacesTableView`	Yes	Yes
IP Addresses	`allIPAddressTableView`	Yes	Yes
IP Subnets	`allIPSubnetsTableView`	No	No
VLANs	`allVlansTableView`	No	No
Cards	`allCardsTableView`	No	No
Ports	`allPortsTableView`	No	No
Nodes by Management Server	`nodesByNNMiManagementServerTableView`	No	No
Custom Nodes	`customNodeTableView`	Yes	No
Custom Interfaces	`customInterfaceTableView`	Yes	Yes
Custom IP Addresses	`customIPAddressTableView`	Yes	Yes
MIB Variables	`mibVariablesTableView`	No	No
Card Redundancy Groups	`allCardRedundancyGroupsTableView`	No	No
Router Redundancy Groups	`routerRedundancyGroupsTableView`	No	No
Node Groups	`nodeGroupsTableView`	No	No
Interface Groups	`interfaceGroupsTableView`	No	No
Management Stations (6.x/7.x)	`allManagementStationsTableView`	No	No

The following are optional filter parameters:

```
http://<serverName>:<portNumber>/nnm/launch?cmd= showView&view=
<x>&nodegroup= <Name>
```

```
http://<serverName>:<portNumber>/nnm/launch?cmd= showView&view=
<x>&interfacegroup= <Name>
```

> **Note:** If you copy/paste this URL, remove the spaces that were added for line-ending purposes in the documentation.

Filter by Node Group (launched Incident, Node, Interface, and IP Address views)

Attribute	Values
nodegroup	The *case-sensitive* Name attribute value of the Node Group to use as a filter for this view. **Note:** The Node Group name is translated. If your team shares NNMi within multiple locales, use nodegroupid or nodegroupuuid. If the Name value includes space characters, consider replacing the space character in your URL statement (see "W3C Rules for URLs" on page 1490).
nodegroupid	The `id` is the Unique Object Identifier (unique across the entire NNMi database). Provide the `id` of the Node Group to use as a filter for this view. This attribute value is not visible on any form within the console. To find this value, use the nnmconfigexport.ovpl command-line tool to export an XML file that includes the `id` attribute value for each object instance.
nodegroupuuid	The `uuid` is the Universally Unique Object Identifier (unique across all databases). Provide the `uuid` of the Node Group to use as a filter for this view. This attribute value is not visible on any form within the console. To find this value, use the nnmconfigexport.ovpl command-line tool to export an XML file that includes the `uuid` attribute value for each object instance.

Filter by Interface Group (launched Interface and IP Address views)

Attribute	Values
ifgroup	The *case-sensitive* Name attribute value of the Interface Group to use as a filter for this view. **Note:** The Interface Group name is translated. If your team shares NNMi within multiple locales, use ifgroupid or ifgroupuuid. If the Name value includes space characters, consider replacing the space character in your URL statement (see "W3C Rules for URLs" on page 1490).
ifgroupid	The `id` is the Unique Object Identifier (unique across the entire NNMi database). Provide the `id` of the Interface Group to use as a filter for this view. This attribute value is not visible on any form within the console. To find this value, use the nnmconfigexport.ovpl command-line tool to export an XML file that includes the `id` attribute value for each object instance.

Filter by Interface Group (launched Interface and IP Address views), continued

Attribute	Values
ifgroupuuid	The `uuid` is the Universally Unique Object Identifier (unique across all databases). Provide the `uuid` of the Interface Group to use as a filter for this view. This attribute value is not visible on any form within the console. To find this value, use the nnmconfigexport.ovpl command-line tool to export an XML file that includes the `uuid` attribute value for each object instance.

The following are optional parameters:

```
http://<serverName>:<portNumber>/nnm/launch?cmd= showView&view=
<x>&menus= <true/false>&newWindow= = <true/false>&envattrs= <name1=
value>;<name2= value>
```

> **Note:** If you copy/paste this URL, remove the spaces that were added for line-ending purposes in the documentation.

Attributes for Launched Views

Attribute	Values
menus	true = Show the view menus and the ☒ Close button. If not specified, the default is true. false = Hide the view menus and the ☒ Close button to save space in the view.
newWindow	true = Display the view in a new browser window. This new window does not display the browser-specific menu bars. The browser-specific menu bars are hidden to provide the maximum amount of room for the requested view. false = Display the view within the current browser window (if not specified, the default is false).
readonly	true = Prevents the user from doing either of the following: • Open any forms from the view • Manipulate any objects in the view (for example, delete an object) false = Enables the user to do either of the following: • Open any forms from the view • Manipulate any objects in the view (for example, delete an object)
envattrs	Use Environment Attributes (`envattrs`) to pass `<name=value>` pairs *from an external application* to NNMi. Environment Attributes are session-specific and stored in memory (not in the NNMi database). The name-value pairs can be any arbitrary string (as required by the external program).

Attributes for Launched Views , continued

Attribute	Values
	For example, let’s assume that NNMi was launched from an application that wants to preserve a session ID (123) and object attribute (node25) as follows: `http://<yourServerName/nnm?cmd= showView&objtype= Node&envattrs= com.my.sessionId= 123;com.my.objectName= node25` **Note:** See "Pass Environment Attributes" on page 1493 for information about how to retrieve these Environment Attributes (`envattrs`) `<name=value>` pairs from NNMi, and pass them back *to the originating external application*.

Launch a Management Mode Workspace Views

> **Note:** If you copy/paste this URL, remove the spaces that were added for line-ending purposes in the documentation.

```
http://<serverName>:<portNumber>/nnm/launch?cmd= showView&view= <x>
```

> **Note:** If the NNMi Web server uses the https protocol, use `https` instead of `http`. See the "Working with Certificates for NNMi" chapter in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.

<serverName> = the fully-qualified domain name of the NNMi management server (values allowed here are determined by the *Enable URL Redirect* setting in User Interface Configuration, see "Configuring the NNMi User Interface" on page 467)

<portNumber> = the NNMi HTTP port number

For a quick-reference list of all URL choices for launching NNMi, see **Help** → **Documentation Library** → **Integrate NNMi Elsewhere with URLs**. The Integrate NNMi Elsewhere with URLs page also contains a link to a list of sample URLs that you can copy/paste as a starting point.

> **Note**: If you are using Mozilla Firefox, click here for more information.
>
> Due to the Mozilla Firefox security implementation, launching an NNMi view with a URL has the following limitations:
>
> - The ☒ Close button does not work.
> - The File → Close menu item does not work.
>
> Use the browser buttons to close the view.

Management Mode Workspace Views

View Name	*x* = View ID	Node Filter	Interface Filter
Unmanaged[1] Nodes	`unManagedNodeTableView`	Yes	No
Unmanaged[2] Interfaces	`unManagedInterfaceTableView`	Yes	Yes
Unmanaged[3] IP Addresses	`unManagedIPAddressTableView`	Yes	Yes
Unmanaged[4] Cards	`unManagedCardTableView`	Yes	No
Unmanaged[5] Node Components	`unManagedNodeComponentTableView`	Yes	No

The following are optional filter parameters:

```
http://<serverName>:<portNumber>/nnm/launch?cmd= showView&view=
<x>&nodegroup= <Name>
```

```
http://<serverName>:<portNumber>/nnm/launch?cmd= showView&view=
<x>&ifgroup= <Name>
```

Note: If you copy/paste this URL, remove the spaces that were added for line-ending purposes in the documentation.

Filter by Node Group (launched Incident, Node, Interface, and IP Address views)

Attribute	Values
nodegroup	The *case-sensitive* Name attribute value of the Node Group to use as a filter for this view. **Note:** The Node Group name is translated. If your team shares NNMi within multiple locales, use nodegroupid or nodegroupuuid. If the Name value includes space characters, consider replacing the space character in your URL statement (see "W3C Rules for URLs" on page 1490).
nodegroupid	The `id` is the Unique Object Identifier (unique across the entire NNMi database). Provide the `id` of the Node Group to use as a filter for this view. This attribute value is not visible on any form within the console. To find this value, use the nnmconfigexport.ovpl command-line tool to export an XML file that includes the `id` attribute value for each object instance.

[1] Indicates the Management Mode is "Not Managed" or "Out of Service".
[2] Indicates the Management Mode is "Not Managed" or "Out of Service".
[3] Indicates the Management Mode is "Not Managed" or "Out of Service".
[4] Indicates the Management Mode is "Not Managed" or "Out of Service".
[5] Indicates the Management Mode is "Not Managed" or "Out of Service".

Filter by Node Group (launched Incident, Node, Interface, and IP Address views), continued

Attribute	Values
nodegroupuuid	The `uuid` is the Universally Unique Object Identifier (unique across all databases). Provide the `uuid` of the Node Group to use as a filter for this view. This attribute value is not visible on any form within the console. To find this value, use the nnmconfigexport.ovpl command-line tool to export an XML file that includes the `uuid` attribute value for each object instance.

Filter by Interface Group (launched Interface and IP Address views)

Attribute	Values
ifgroup	The *case-sensitive* Name attribute value of the Interface Group to use as a filter for this view. **Note:** The Interface Group name is translated. If your team shares NNMi within multiple locales, use ifgroupid or ifgroupuuid. If the Name value includes space characters, consider replacing the space character in your URL statement (see "W3C Rules for URLs" on page 1490).
ifgroupid	The `id` is the Unique Object Identifier (unique across the entire NNMi database). Provide the `id` of the Interface Group to use as a filter for this view. This attribute value is not visible on any form within the console. To find this value, use the nnmconfigexport.ovpl command-line tool to export an XML file that includes the `id` attribute value for each object instance.
ifgroupuuid	The `uuid` is the Universally Unique Object Identifier (unique across all databases). Provide the `uuid` of the Interface Group to use as a filter for this view. This attribute value is not visible on any form within the console. To find this value, use the nnmconfigexport.ovpl command-line tool to export an XML file that includes the `uuid` attribute value for each object instance.

```
http://<serverName>:<portNumber>/nnm/launch?cmd= showView&view=
<x>&menus= <true/false>&newWindow= = <true/false>&envattrs= <name1=
value>;<name2= value>
```

Note: If you copy/paste this URL, remove the spaces that were added for line-ending purposes in the documentation.

Attributes for Launched Views

Attribute	Values
menus	true = Show the view menus and the ☒ Close button. If not specified, the default is true.

Attributes for Launched Views , continued

Attribute	Values
	false = Hide the view menus and the Close button to save space in the view.
newWindow	true = Display the view in a new browser window. This new window does not display the browser-specific menu bars. The browser-specific menu bars are hidden to provide the maximum amount of room for the requested view. false = Display the view within the current browser window (if not specified, the default is false).
readonly	true = Prevents the user from doing either of the following: • Open any forms from the view • Manipulate any objects in the view (for example, delete an object) false = Enables the user to do either of the following: • Open any forms from the view • Manipulate any objects in the view (for example, delete an object)
envattrs	Use Environment Attributes (`envattrs`) to pass `<`*`name=value`*`>` pairs *from an external application* to NNMi. Environment Attributes are session-specific and stored in memory (not in the NNMi database). The name-value pairs can be any arbitrary string (as required by the external program). For example, let’s assume that NNMi was launched from an application that wants to preserve a session ID (123) and object attribute (node25) as follows: `http://<yourServerName/nnm?cmd= showView&objtype= Node&envattrs= com.my.sessionId= 123;com.my.objectName= node25` **Note:** See "Pass Environment Attributes" on page 1493 for information about how to retrieve these Environment Attributes (`envattrs`) `<`*`name=value`*`>` pairs from NNMi, and pass them back *to the originating external application*.

Launch a Configuration Workspace View

Configuration workspaces require that the user be assigned to the **Administrative** role.

> **Note:** If you copy/paste this URL, remove the spaces that were added for line-ending purposes in the documentation.

```
http://<serverName>:<portNumber>/nnm/launch?cmd= showView&view= <x>
```

> **Note:** If the NNMi Web server uses the https protocol, use `https` instead of `http`. See the "Working with Certificates for NNMi" chapter in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.

<*serverName*> = the fully-qualified domain name of the NNMi management server (values allowed here are determined by the *Enable URL Redirect* setting in User Interface Configuration, see "Configuring the NNMi User Interface" on page 467)

<*portNumber*> = the NNMi HTTP port number

For a quick-reference list of all URL choices for launching NNMi, see **Help** → **Documentation Library** → **Integrate NNMi Elsewhere with URLs**. The Integrate NNMi Elsewhere with URLs page also contains a link to a list of sample URLs that you can copy/paste as a starting point.

> **Note**: If you are using Mozilla Firefox, click here for more information.
>
> Due to the Mozilla Firefox security implementation, launching an NNMi view with a URL has the following limitations:
>
> - The ☒ Close button does not work.
> - The File → Close menu item does not work.
>
> Use the browser buttons to close the view.

Configuration Workspace Views

View Name	*x* = View ID
Node Groups	`nodeGroupsTableView`
Interface Groups	`interfaceGroupsTableView`
ifTypes	`allIfTypesTableView`
Device Profiles	`allDeviceProfilesTableView`
Loaded MIBs	`loadedMibsTableView`
MIB Expressions	`mibExpressionsTableView`
RAMS Servers	`ramsServerTableView`
Management Stations (6.x/7.x)	`allManagementStationsTableView`

The following are optional parameters:

```
http://<serverName>:<portNumber>/nnm/launch?cmd= showView&view=
<x>&menus= <true/false>&newWindow= = <true/false>&envattrs= <name1=
value>;<name2= value>
```

> **Note:** If you copy/paste this URL, remove the spaces that were added for line-ending purposes in the documentation.

Attributes for Launched Views

Attribute	Values
menus	true = Show the view menus and the ☒ Close button. If not specified, the default is true. false = Hide the view menus and the ☒ Close button to save space in the view.
newWindow	true = Display the view in a new browser window. This new window does not display the browser-specific menu bars. The browser-specific menu bars are hidden to provide the maximum amount of room for the requested view. false = Display the view within the current browser window (if not specified, the default is false).
readonly	true = Prevents the user from doing either of the following: • Open any forms from the view • Manipulate any objects in the view (for example, delete an object) false = Enables the user to do either of the following: • Open any forms from the view • Manipulate any objects in the view (for example, delete an object)
envattrs	Use Environment Attributes (`envattrs`) to pass `<`*`name=value`*`>` pairs *from an external application* to NNMi. Environment Attributes are session-specific and stored in memory (not in the NNMi database). The name-value pairs can be any arbitrary string (as required by the external program). For example, let’s assume that NNMi was launched from an application that wants to preserve a session ID (123) and object attribute (node25) as follows: `http://<yourServerName/nnm?cmd= showView&objtype= Node&envattrs= com.my.sessionId= 123;com.my.objectName= node25` **Note:** See "Pass Environment Attributes" on page 1493 for information about how to retrieve these Environment Attributes (`envattrs`) `<`*`name=value`*`>` pairs from NNMi, and pass them back *to the originating external application*.

Launch a Form (showForm)

To launch a particular form, use the following URL:

`http://<`*`serverName`*`>:<`*`portNumber`*`>/nnm/launch?`**`cmd=showForm`**`...`

> **Note:** If the NNMi Web server uses the https protocol, use `https` instead of `http`. See the "Working with Certificates for NNMi" chapter in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.

<serverName> = the fully-qualified domain name of the NNMi management server (values allowed here are determined by the *Enable URL Redirect* setting in User Interface Configuration, see "Configuring the NNMi User Interface" on page 467)

<portNumber> = the NNMi HTTP port number

For a quick-reference list of all URL choices for launching NNMi, see **Help** → **Documentation Library** → **Integrate NNMi Elsewhere with URLs**. The Integrate NNMi Elsewhere with URLs page also contains a link to a list of sample URLs that you can copy/paste as a starting point.

Launch a form to see information about a particular node, interface, address, subnet, or incident. In the URL string, you must include one or more attributes that enable NNMi to find a specific object. If more than one object meets the criteria, NNMi opens the first one found. When designating more than one attribute, separate each with a semicolon character.

"Launch a Node Form" below

"Launch an Interface Form" on page 1542

"Launch an IP Address Form" on page 1544

"Launch a Subnet Form" on page 1545

"Launch an Incident Form" on page 1547

"Launch a Node Group Form" on page 1549

"Launch a Configuration Form" on page 1551

Launch a Node Form

> **Note:** If you copy/paste this URL, remove the spaces that were added for line-ending purposes in the documentation.

```
http://<serverName>:<portNumber>/nnm/launch?cmd= showForm&objtype=
Node&nodename= <x>
```

```
http://<serverName>:<portNumber>/nnm/launch?cmd= showForm&objtype=
Node&objattrs= name= <x>
```

```
http://<serverName>:<portNumber>/nnm/launch?cmd= showForm&objtype=
Node&objattrs= hostname= <x>
```

```
http://<serverName>:<portNumber>/nnm/launch?cmd= showForm&objtype=
Node&objattrs= snmpAgent.agentSettings.managementAddress= <x>
```

```
http://<serverName>:<portNumber>/nnm/launch?cmd= showForm&objtype=
Node&objattrs= systemName= <x>
```

> **Note:** If the NNMi Web server uses the https protocol, use `https` instead of `http`. See the "Working with Certificates for NNMi" chapter in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.

<serverName> = the fully-qualified domain name of the NNMi management server (values allowed here are determined by the *Enable URL Redirect* setting in User Interface Configuration, see "Configuring the NNMi User Interface" on page 467)

<portNumber> = the NNMi HTTP port number

For a quick-reference list of all URL choices for launching NNMi, see **Help** → **Documentation Library** → **Integrate NNMi Elsewhere with URLs**. The Integrate NNMi Elsewhere with URLs page also contains a link to a list of sample URLs that you can copy/paste as a starting point.

> **Note**: If you are using Mozilla Firefox, click here for more information.
>
> Due to the Mozilla Firefox security implementation, launching an NNMi form with a URL has the following limitations:
>
> - The Close button does not work.
> - The File → Close menu item does not work.
> - The Save and Close button saves data, but does not close the window.
> - The File → Save and Close menu item saves data, but does not close the window.
>
> Use the browser buttons to close the form.

Node Form Attributes

Attribute	Values
nodename	Provide the node's DNS hostname (full or short) or IP address. If you use this attribute, NNMi tries to match the string you provide by following this procedure: • Check the value of the Hostname (*case-sensitive*) on the Node form. • Check the values in the Address column of the table on the Node form, Addresses tab, • Check the value of the System Name field on the in the Node form, General tab. • Check the value in the Name field on the Node form.
name	The Name attribute value from the Node form.
hostname	The *case-sensitive* Hostname attribute value from the Node form of the discovered node must match what is entered here. NNMi follows a set of rules to dynamically generate the value

Node Form Attributes , continued

Attribute	Values
	stored in the NNMi database for each Node's Hostname. Click here for details. • If the Node supports SNMP, NNMi requests the Hostname using the IP Address of the associated SNMP agent (the Management Address attribute value on the Node form). When the NNMi administrator chooses **Enable SNMP Address Rediscovery** ☑ in the Communication Configuration: ▪ If the SNMP Agent does not respond, NNMi checks for another Management Address to request the Hostname, and the Hostname could change. ▪ If the SNMP Agent associated with the node changes, the Management Address and Hostname could change. When the NNMi administrator disables **Enable SNMP Address Rediscovery** □ in the Communication Configuration: ▪ If the SNMP Agent does not respond, NNMi uses the previously gathered Management Address attribute value to request the Hostname. ▪ If the SNMP Agent associated with the node changes, NNMi uses the previously gathered Management Address attribute value to request the Hostname. • If the Node does not support SNMP, no Management Address is available. NNMi requests a Hostname starting with the lowest IP Address associated with the node (a Discovery Seed value or an IP address value gathered from a neighboring device). NNMi uses the first Hostname provided. The Hostname might change during a future discovery cycle. **Note:** NNMi administrators can use NNMi property file settings to change the way NNMi determines Hostname values: • `nms-topology.properties` file settings: If DNS is the source of the Node's Hostname, there are three choices. By default NNMi uses the exact Hostname from your network configuration. It is possible to change NNMi behavior to convert Hostnames to all uppercase or all lowercase. See the "Modifying NNMi Normalization Properties" section of the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`. • `nms-disco.properties` file settings:

Node Form Attributes , continued

Attribute	Values
	The Hostname is either requested from the Node's lowest loopback interface IP address that resolves to a Hostname or requested from the Node's designated Management Address (SNMP agent address). With either choice, when no IP address resolves to a Hostname, the IP address itself becomes the Hostname. See the "Maintaining NNMi" chapter of the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.
snmpAgent.agentSettings.managementAddress	The Management Address attribute value from the SNMP Agent form of the agent assigned to the specified node. The value is an IP address.
systemName	System Name attribute value from the Node form, General tab.

The following are optional filter parameters:

```
http://<serverName>:<portNumber>/nnm/launch?cmd= showForm&objtype=
Node&nodename= <x>&menus= <true/false>&envattrs= <name1=
value>;<name2= value>
```

Note: If you copy/paste this URL, remove the spaces that were added for line-ending purposes in the documentation.

Attributes for Launched Forms

Attribute	Values
menus	true = Show the view menus and the ☒ Close button. If not specified, the default is true. false = Hide the view menus and the ☒ Close button to save space in the view.
envattrs	Use Environment Attributes (`envattrs`) to pass `<name=value>` pairs *from an external application* to NNMi. Environment Attributes are session-specific and stored in memory (not in the NNMi database). The name-value pairs can be any arbitrary string (as required by the external program). For example, let’s assume that NNMi was launched from an application that wants to preserve a session ID (123) and object attribute (node25) as follows: `http://<yourServerName/nnm?cmd= showView&objtype= Node&envattrs= com.my.sessionId= 123;com.my.objectName= node25`

Attributes for Launched Forms , continued

Attribute	Values
	Note: See "Pass Environment Attributes" on page 1493 for information about how to retrieve these Environment Attributes (envattrs) <*name*=*value*> pairs from NNMi, and pass them back *to the originating external application*.

Launch an Interface Form

Note: If you copy/paste this URL, remove the spaces that were added for line-ending purposes in the documentation.

http://<*serverName*>:<*portNumber*>/nnm/launch?cmd= **showForm&objtype= Interface&objattrs= hostedOn.hostname= <*x*>;name= <y>**

http://<*serverName*>:<*portNumber*>/nnm/launch?cmd= showForm&objtype= Interface&objattrs= hostedOn.hostname= <*x*>;**ifName= <y>**

http://<*serverName*>:<*portNumber*>/nnm/launch?cmd= showForm&objtype= Interface&objattrs= hostedOn.hostname= <*x*>;**ifAlias= <y>**

http://<*serverName*>:<*portNumber*>/nnm/launch?cmd= showForm&objtype= Interface&objattrs= hostedOn.hostname= <*x*>;**ifIndex= <y>**

Note: If the NNMi Web server uses the https protocol, use `https` instead of `http`. See the "Working with Certificates for NNMi" chapter in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.

<*serverName*> = the fully-qualified domain name of the NNMi management server (values allowed here are determined by the *Enable URL Redirect* setting in User Interface Configuration, see "Configuring the NNMi User Interface" on page 467)

<*portNumber*> = the NNMi HTTP port number

For a quick-reference list of all URL choices for launching NNMi, see **Help → Documentation Library → Integrate NNMi Elsewhere with URLs**. The Integrate NNMi Elsewhere with URLs page also contains a link to a list of sample URLs that you can copy/paste as a starting point.

Note: If you are using Mozilla Firefox, click here for more information.

Due to the Mozilla Firefox security implementation, launching an NNMi form with a URL has the following limitations:

- The ☒ Close button does not work.
- The File → Close menu item does not work.

- The Save and Close button saves data, but does not close the window.
- The File → Save and Close menu item saves data, but does not close the window.

Use the browser buttons to close the form.

Interface Form Attributes

Attribute	Values
hostedOn.hostname	The *case-sensitive* Hostname of the Node in which the interface resides. This is the Hostname attribute value from the associated Node form.
name	The Name attribute value from the Interface form.
ifName	The IfName attribute value from the Interface form.
ifAlias	The IfAlias attribute value from the Interface form.
ifIndex	The IfIndex attribute value from the Interface form.

The following are optional filter parameters:

```
http://<serverName>:<portNumber>/nnm/launch?cmd= showForm&objtype=
Interface&objattrs= hostedOn.hostname= <x>;name= <y>&menus=
<true/false>&envattrs= <name1= value>;<name2= value>
```

Note: If you copy/paste this URL, remove the spaces that were added for line-ending purposes in the documentation.

Attributes for Launched Forms

Attribute	Values
menus	true = Show the view menus and the Close button. If not specified, the default is true. false = Hide the view menus and the Close button to save space in the view.
envattrs	Use Environment Attributes (`envattrs`) to pass `<name=value>` pairs *from an external application* to NNMi. Environment Attributes are session-specific and stored in memory (not in the NNMi database). The name-value pairs can be any arbitrary string (as required by the external program). For example, let's assume that NNMi was launched from an application that wants to preserve a session ID (123) and object attribute (node25) as follows: `http://<yourServerName/nnm?cmd= showView&objtype= Node&envattrs= com.my.sessionId= 123;com.my.objectName= node25` **Note:** See "Pass Environment Attributes" on page 1493 for information about how to retrieve these Environment Attributes (`envattrs`) `<name=value>` pairs from NNMi, and pass them back *to the originating external application*.

Launch an IP Address Form

> **Note:** If you copy/paste this URL, remove the spaces that were added for line-ending purposes in the documentation.

```
http://<serverName>:<portNumber>/nnm/launch?cmd= showForm&objtype=
IPAddress&objattrs= value= <y>
```

> **Note:** If the NNMi Web server uses the https protocol, use `https` instead of `http`. See the "Working with Certificates for NNMi" chapter in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.

<*serverName*> = the fully-qualified domain name of the NNMi management server (values allowed here are determined by the *Enable URL Redirect* setting in User Interface Configuration, see "Configuring the NNMi User Interface" on page 467)

<*portNumber*> = the NNMi HTTP port number

For a quick-reference list of all URL choices for launching NNMi, see **Help** → **Documentation Library** → **Integrate NNMi Elsewhere with URLs**. The Integrate NNMi Elsewhere with URLs page also contains a link to a list of sample URLs that you can copy/paste as a starting point.

> **Note**: If you are using Mozilla Firefox, click here for more information.
>
> Due to the Mozilla Firefox security implementation, launching an NNMi form with a URL has the following limitations:
>
> - The Close button does not work.
> - The File → Close menu item does not work.
> - The Save and Close button saves data, but does not close the window.
> - The File → Save and Close menu item saves data, but does not close the window.
>
> Use the browser buttons to close the form.

IP Address Form Attributes

Attribute	Values
value	The Address attribute value from the IP Address form.

The following are optional filter parameters:

```
http://<serverName>:<portNumber>/nnm/launch?cmd= showForm&objtype=
IPAddress&objattrs= value= <y>&menus= <true/false>&envattrs= <name1=
value>;<name2= value>
```

> **Note:** If you copy/paste this URL, remove the spaces that were added for line-ending purposes

in the documentation.

Attributes for Launched Forms

Attribute	Values
menus	true = Show the view menus and the Close button. If not specified, the default is true. false = Hide the view menus and the Close button to save space in the view.
envattrs	Use Environment Attributes (`envattrs`) to pass <*name=value*> pairs *from an external application* to NNMi. Environment Attributes are session-specific and stored in memory (not in the NNMi database). The name-value pairs can be any arbitrary string (as required by the external program). For example, let's assume that NNMi was launched from an application that wants to preserve a session ID (123) and object attribute (node25) as follows: `http://<yourServerName/nnm?cmd= showView&objtype= Node&envattrs= com.my.sessionId= 123;com.my.objectName= node25` **Note:** See "Pass Environment Attributes" on page 1493 for information about how to retrieve these Environment Attributes (`envattrs`) <*name=value*> pairs from NNMi, and pass them back *to the originating external application*.

Launch a Subnet Form

Note: If you copy/paste this URL, remove the spaces that were added for line-ending purposes in the documentation.

http://<*serverName*>:<*portNumber*>/nnm/launch?cmd= **showForm&objtype=
IPSubnet&objattrs= name= <*x*>**

http://<*serverName*>:<*portNumber*>/nnm/launch?cmd= showForm&objtype=
IPSubnet&**objattrs= prefix= <*x*>**

http://<*serverName*>:<*portNumber*>/nnm/launch?cmd= showForm&objtype=
IPSubnet&**objattrs= prefix= <*x*>;prefixLength= <*y*>**

Note: If the NNMi Web server uses the https protocol, use `https` instead of `http`. See the "Working with Certificates for NNMi" chapter in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.

<*serverName*> = the fully-qualified domain name of the NNMi management server (values allowed here are determined by the *Enable URL Redirect* setting in User Interface Configuration, see "Configuring the NNMi User Interface" on page 467)

`<portNumber>` = the NNMi HTTP port number

For a quick-reference list of all URL choices for launching NNMi, see **Help → Documentation Library → Integrate NNMi Elsewhere with URLs**. The Integrate NNMi Elsewhere with URLs page also contains a link to a list of sample URLs that you can copy/paste as a starting point.

> **Note**: If you are using Mozilla Firefox, click here for more information.
>
> Due to the Mozilla Firefox security implementation, launching an NNMi form with a URL has the following limitations:
>
> - The Close button does not work.
> - The File → Close menu item does not work.
> - The Save and Close button saves data, but does not close the window.
> - The File → Save and Close menu item saves data, but does not close the window.
>
> Use the browser buttons to close the form.

IP Subnet Form Attributes

Attribute	Values
name	The *case-sensitive* Name attribute value from the IP Subnet form.
prefix	The Prefix attribute value from the IP Subnet form.
prefixLength	The Prefix Length attribute value from the IP Subnet form.

The following are optional filter parameters:

```
http://<serverName>:<portNumber>/nnm/launch?cmd= showForm&objtype=
IPSubnet&objattrs= name= <x>&menus= <true/false>&envattrs= <name1=
value>;<name2= value>
```

> **Note:** If you copy/paste this URL, remove the spaces that were added for line-ending purposes in the documentation.

Attributes for Launched Forms

Attribute	Values
menus	true = Show the view menus and the Close button. If not specified, the default is true. false = Hide the view menus and the Close button to save space in the view.
envattrs	Use Environment Attributes (`envattrs`) to pass `<name=value>` pairs *from an external application* to NNMi. Environment Attributes are session-specific and stored in memory (not in the NNMi database). The name-value pairs can be any arbitrary string (as required by the external program). For example, let's assume that NNMi was launched from an application that wants to preserve a session ID (123) and object attribute (node25) as follows:

Attributes for Launched Forms , continued

Attribute	Values
	http://<yourServerName/nnm?cmd= showView&objtype= Node&envattrs= com.my.sessionId= 123;com.my.objectName= node25 **Note:** See "Pass Environment Attributes" on page 1493 for information about how to retrieve these Environment Attributes (envattrs) <*name=value*> pairs from NNMi, and pass them back *to the originating external application*.

Launch an Incident Form

Note: If you copy/paste this URL, remove the spaces that were added for line-ending purposes in the documentation.

http://<*serverName*>:<*portNumber*>/nnm/launch?cmd= **showForm&objtype= Incident&objid= <*x*>**

http://<*serverName*>:<*portNumber*>/nnm/launch?cmd= **showForm&objtype= Incident&objuuid= <*x*>**

Note: If the NNMi Web server uses the https protocol, use https instead of http. See the "Working with Certificates for NNMi" chapter in the *HP Network Node Manager i Software Deployment Reference*, which is available at: http://h20230.www2.hp.com/selfsolve/manuals.

<*serverName*> = the fully-qualified domain name of the NNMi management server (values allowed here are determined by the *Enable URL Redirect* setting in User Interface Configuration, see "Configuring the NNMi User Interface" on page 467)

<*portNumber*> = the NNMi HTTP port number

For a quick-reference list of all URL choices for launching NNMi, see **Help → Documentation Library → Integrate NNMi Elsewhere with URLs**. The Integrate NNMi Elsewhere with URLs page also contains a link to a list of sample URLs that you can copy/paste as a starting point.

Note: If you are using Mozilla Firefox, click here for more information.

Due to the Mozilla Firefox security implementation, launching an NNMi form with a URL has the following limitations:

- The Close button does not work.
- The File → Close menu item does not work.
- The Save and Close button saves data, but does not close the window.
- The File → Save and Close menu item saves data, but does not close the window.

Use the browser buttons to close the form.

Individual incident objects must be identified by their *database unique identifiers*.

Incident Attributes

Attribute	Values
objid	The Unique Object Identifier (unique per object type in the NNMi database). NNMi displays the `id` attribute value on the object form's Registration tab.
objuuid	The Universally Unique Object Identifier (unique across all databases). NNMi displays the `uuid` attribute value on the object form's Registration tab.

The following are optional filter parameters:

```
http://<serverName>:<portNumber>/nnm/launch?cmd=
showForm&showForm&objtype= Incident&objid= <x>&menus=
<true/false>&envattrs= <name1= value>;<name2= value>
```

> **Note:** If you copy/paste this URL, remove the spaces that were added for line-ending purposes in the documentation.

Attributes for Launched Forms

Attribute	Values
menus	true = Show the view menus and the ☒ Close button. If not specified, the default is true. false = Hide the view menus and the ☒ Close button to save space in the view.
envattrs	Use Environment Attributes (`envattrs`) to pass `<`*`name=value`*`>` pairs *from an external application* to NNMi. Environment Attributes are session-specific and stored in memory (not in the NNMi database). The name-value pairs can be any arbitrary string (as required by the external program). For example, let’s assume that NNMi was launched from an application that wants to preserve a session ID (123) and object attribute (node25) as follows: `http://<yourServerName/nnm?cmd= showView&objtype= Node&envattrs= com.my.sessionId= 123;com.my.objectName= node25` **Note:** See "Pass Environment Attributes" on page 1493 for information about how to retrieve these Environment Attributes (`envattrs`) `<`*`name=value`*`>` pairs from NNMi, and pass them back *to the originating external application*.

Launch a Node Group Form

> **Note:** If you copy/paste this URL, remove the spaces that were added for line-ending purposes in the documentation.

http://<*serverName*>:<*portNumber*>/nnm/launch?cmd= **showForm&objtype= NodeGroup&name= <*y*>**

http://<*serverName*>:<*portNumber*>/nnm/launch?cmd= showForm&objtype= NodeGroup**&nodegroupid= <*y*>**

http://<*serverName*>:<*portNumber*>/nnm/launch?cmd= showForm&objtype= NodeGroup**&nodegroupuuid= <*y*>**

> **Note:** If the NNMi Web server uses the https protocol, use `https` instead of `http`. See the "Working with Certificates for NNMi" chapter in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.

<*serverName*> = the fully-qualified domain name of the NNMi management server (values allowed here are determined by the *Enable URL Redirect* setting in User Interface Configuration, see "Configuring the NNMi User Interface" on page 467)

<*portNumber*> = the NNMi HTTP port number

For a quick-reference list of all URL choices for launching NNMi, see **Help → Documentation Library → Integrate NNMi Elsewhere with URLs**. The Integrate NNMi Elsewhere with URLs page also contains a link to a list of sample URLs that you can copy/paste as a starting point.

> **Note**: If you are using Mozilla Firefox, click here for more information.
>
> Due to the Mozilla Firefox security implementation, launching an NNMi form with a URL has the following limitations:
>
> - The Close button does not work.
> - The File → Close menu item does not work.
> - The Save and Close button saves data, but does not close the window.
> - The File → Save and Close menu item saves data, but does not close the window.
>
> Use the browser buttons to close the form.

Node Group Form Attributes

Attribute	Values
name	The *case-sensitive* Name attribute value from the Node Group form. **Note**: The Node Group name is translated. If your team shares NNMi within multiple locales, use nodegroupid or nodegroupuuid.

Node Group Form Attributes, continued

Attribute	Values
	If the Name value includes space characters, consider replacing the space character in your URL statement (see "W3C Rules for URLs" on page 1490).
nodegroupid	The `id` is the Unique Object Identifier (unique per object type in the NNMi database). Provide the `id` of the Node Group to use as a filter for this view. This attribute value is not visible on any form within the console. To find this value, use the nnmconfigexport.ovpl command-line tool to export an XML file that includes the `id` attribute value for each object instance.
nodegroupuuid	The `uuid` is the Universally Unique Object Identifier (unique across all databases). Provide the `uuid` of the Node Group to use as a filter for this view. This attribute value is not visible on any form within the console. To find this value, use the nnmconfigexport.ovpl command-line tool to export an XML file that includes the `uuid` attribute value for each object instance.

The following are optional filter parameters:

```
http://<serverName>:<portNumber>/nnm/launch?cmd= showForm&objtype=
NodeGroup&name= <y>&menus= <true/false>&envattrs= <name1=
value>;<name2= value>
```

> **Note:** If you copy/paste this URL, remove the spaces that were added for line-ending purposes in the documentation.

Attributes for Launched Forms

Attribute	Values
menus	true = Show the view menus and the ☒ Close button. If not specified, the default is true. false = Hide the view menus and the ☒ Close button to save space in the view.
envattrs	Use Environment Attributes (`envattrs`) to pass `<name=value>` pairs *from an external application* to NNMi. Environment Attributes are session-specific and stored in memory (not in the NNMi database). The name-value pairs can be any arbitrary string (as required by the external program). For example, let's assume that NNMi was launched from an application that wants to preserve a session ID (123) and object attribute (node25) as follows: `http://<yourServerName/nnm?cmd= showView&objtype= Node&envattrs= com.my.sessionId= 123;com.my.objectName= node25` **Note:** See "Pass Environment Attributes" on page 1493 for information about how to retrieve these Environment Attributes (`envattrs`) `<name=value>` pairs from NNMi, and pass them back *to the originating external application*.

Launch a Configuration Form

Configuration forms require that the user be assigned to the **Administrative** role.

> **Note:** If you copy/paste this URL, remove the spaces that were added for line-ending purposes in the documentation.

http://<*serverName*>:<*portNumber*>/nnm/launch?cmd= **showConfigForm&name=<*y*>**

> **Note:** If the NNMi Web server uses the https protocol, use `https` instead of `http`. See the "Working with Certificates for NNMi" chapter in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.

<*serverName*> = the fully-qualified domain name of the NNMi management server (values allowed here are determined by the *Enable URL Redirect* setting in User Interface Configuration, see "Configuring the NNMi User Interface" on page 467)

<*portNumber*> = the NNMi HTTP port number

For a quick-reference list of all URL choices for launching NNMi, see **Help → Documentation Library → Integrate NNMi Elsewhere with URLs**. The Integrate NNMi Elsewhere with URLs page also contains a link to a list of sample URLs that you can copy/paste as a starting point.

> **Note**: If you are using Mozilla Firefox, click here for more information.
>
> Due to the Mozilla Firefox security implementation, launching an NNMi form with a URL has the following limitations:
>
> - The Close button does not work.
> - The File → Close menu item does not work.
> - The Save and Close button saves data, but does not close the window.
> - The File → Save and Close menu item saves data, but does not close the window.
>
> Use the browser buttons to close the form.

Configuration Form Attributes

Attribute	Values
name	The name attribute value specifies which form: • **customcorrelation** = the Custom Correlation Configuration • **communication** = the Communication Configuration form • **custompoller** = the Custom Poller Configuration form • **discovery** = the Discovery Configuration form

Configuration Form Attributes, continued

Attribute	Values
	• **globalnetworkmanagement** = the Global Network Management form • **monitoring** = the Monitoring Configuration form • **incident** = the Incident Configuration form • **status** = the Status Configuration form • **trap** = the Trap Forwarding Configuration form • **ui** = the User Interface Configuration form

The following are optional filter parameters:

`http://<`*`serverName`*`>:<`*`portNumber`*`>/nnm/launch?cmd= showConfigForm&name=`
`<`*`x`*`>`**`&menus= <`*`true/false`*`>&envattrs= <`*`name1= value`*`>;<`*`name2= value`*`>`**

> **Note:** If you copy/paste this URL, remove the spaces that were added for line-ending purposes in the documentation.

Attributes for Launched Forms

Attribute	Values
menus	true = Show the view menus and the Close button. If not specified, the default is true. false = Hide the view menus and the Close button to save space in the view.
envattrs	Use Environment Attributes (`envattrs`) to pass `<`*`name=value`*`>` pairs *from an external application* to NNMi. Environment Attributes are session-specific and stored in memory (not in the NNMi database). The name-value pairs can be any arbitrary string (as required by the external program). For example, let's assume that NNMi was launched from an application that wants to preserve a session ID (123) and object attribute (node25) as follows: `http://<yourServerName/nnm?cmd= showView&objtype= Node&envattrs= com.my.sessionId= 123;com.my.objectName= node25` **Note:** See "Pass Environment Attributes" on page 1493 for information about how to retrieve these Environment Attributes (`envattrs`) `<`*`name=value`*`>` pairs from NNMi, and pass them back *to the originating external application*.

Launch Menu Items (runTool)

To execute a Launch Action requesting something from NNMi:

```
http://<serverName>:<portNumber>/nnm/launch?cmd=showMenuItem&key=<Menu
ItemKey>[&nodename=<hostname or IP_address>
```

> **Note:** If the NNMi Web server uses the https protocol, use `https` instead of `http`. See the "Working with Certificates for NNMi" chapter in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.

<serverName> = the fully-qualified domain name of the NNMi management server (values allowed here are determined by the *Enable URL Redirect* setting in User Interface Configuration, see "Configuring the NNMi User Interface" on page 467)

<portNumber> = the NNMi HTTP port number

For a quick-reference list of all URL choices for launching NNMi, see **Help → Documentation Library → Integrate NNMi Elsewhere with URLs**. The Integrate NNMi Elsewhere with URLs page also contains a link to a list of sample URLs that you can copy/paste as a starting point.

To execute a Launch Action requesting a script, application, or tool from your environment (not NNMi):

```
http://<
serverName>:<portNumber>/<application>?<yourURLparameter1>=${<attribut
e>}&<yourURLparameter2>=${<attribute>}
```

> **Note:** To extend the NNMi environment with additional applications, you must deploy them into a separate web-server or application-server on the same or different physical server from where the NNMi web-server or application-server is installed. See the *NNMi Developer's Toolkit* for more information.

<serverName> = the appropriate fully-qualified domain name

<portNumber> = the appropriate port number

Provide quick access to NNMi menu items wherever your team needs them:

"Launch the Actions: Communication Configuration Command" on the next page

"Launch the Actions: Configuration Poll Command" on page 1555

"Launch the Actions: Line Graph (showLineGraph)" on page 1556

"Launch the Actions: Monitoring Settings Command" on page 1558

"Launch the Actions: Ping Command" on page 1563

"Launch the Actions: Status Details Command (for Node Groups)" on page 1564

"Launch the Actions: Status Poll Command" on page 1565

"Launch the Actions: Trace Route Command" on page 1566

"Actions: Execute a Launch Action" on page 1567

"Launch the Tools: MIB Browser (showMibBrowser)" on page 1568

"Launch the Tools: NNMi Status Command" on page 1570

"Launch the File: Sign-Out Command" on page 1571

"Launch the Tools: Sign-In/Out Audit Log Command" on page 1571

Launch the Actions: Communication Configuration Command

This URL is equivalent to the **Actions** → **Communication Settings** command in the console.

For a quick-reference list of all URL choices for launching NNMi, see **Help** → **Documentation Library** → **Integrate NNMi Elsewhere with URLs**. The Integrate NNMi Elsewhere with URLs page also contains a link to a list of sample URLs that you can copy/paste as a starting point.

To launch a window that reports the current ICMP and SNMP configuration for a node, use the following URL:

```
http://<serverName>:<portNumber>/nnm/launch?cmd=runTool&tool=commconf
```

> **Note:** If the NNMi Web server uses the https protocol, use `https` instead of `http`. See the "Working with Certificates for NNMi" chapter in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.

`<serverName>` = the fully-qualified domain name of the NNMi management server (values allowed here are determined by the *Enable URL Redirect* setting in User Interface Configuration, see "Configuring the NNMi User Interface" on page 467)

`<portNumber>` = the NNMi HTTP port number

After you specify a node, the real-time results of the ICMP and SNMP configuration report appear.

To launch the real-time results of the ICMP and SNMP configuration report, use the following URL:

```
http://<
serverName
>:<portNumber>/nnm/launch?cmd=runTool&tool=commconf&nodename=<x>
```

```
http://<
serverName
>:<portNumber>/nnm/launch?cmd=runTool&tool=commconf&IPAddress=<x>
```

```
http://<
serverName
>:<portNumber>/nnm/launch?cmd=runTool&tool=commconf&Interface=<x>
```

```
http://<
serverName
>:<portNumber>/nnm/launch?cmd=runTool&tool=commconf&snmpAgent=<x>
```

Communication Configuration Command Attributes

Attribute	Values
nodename	The node's DNS hostname (full or short) or IP address.

Communication Configuration Command Attributes, continued

Attribute	Values
	If you use this attribute, NNMi tries to match the string you provide by following this procedure: • Check the value of the Hostname (*case-sensitive*) on the Node form. • Check the values in the Address column of the table on the Node form, Addresses tab, • Check the value of the System Name field on the in the Node form, General tab. • Check the value in the Name field on the Node form.

Related Topics:

"Troubleshooting Communication Settings" on page 169

Launch the Actions: Configuration Poll Command

This URL is equivalent to the **Actions** → **Polling** → **Configuration Poll** command in the console.

For a quick-reference list of all URL choices for launching NNMi, see **Help** → **Documentation Library** → **Integrate NNMi Elsewhere with URLs**. The Integrate NNMi Elsewhere with URLs page also contains a link to a list of sample URLs that you can copy/paste as a starting point.

To launch a window that reports the current configuration for a node, use the following URL:

```
http://<
serverName>:<portNumber>/nnm/launch?cmd=runTool&tool=configurationpoll
```

> **Note:** If the NNMi Web server uses the https protocol, use `https` instead of `http`. See the "Working with Certificates for NNMi" chapter in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.

<serverName> = the fully-qualified domain name of the NNMi management server (values allowed here are determined by the *Enable URL Redirect* setting in User Interface Configuration, see "Configuring the NNMi User Interface" on page 467)

<portNumber> = the NNMi HTTP port number

After you specify a node, the real-time results of the node's configuration appear.

To launch the real-time results of a node's configuration, use the following URL:

```
http://<
serverName
>:<
portNumber>/nnm/launch?cmd=runTool&tool=configurationpoll&nodename=<x>
```

Configuration Poll Command Attributes

Attribute	Values
nodename	The node's DNS hostname (full or short) or IP address. If you use this attribute, NNMi tries to match the string you provide by following this procedure: • Check the value of the Hostname (*case-sensitive*) on the Node form. • Check the values in the Address column of the table on the Node form, Addresses tab, • Check the value of the System Name field on the in the Node form, General tab. • Check the value in the Name field on the Node form.

Related Topics:

Verify Device Configuration Details

Launch the Actions: Line Graph (showLineGraph)

Use the showLineGraph URL to launch a Line Graph that displays real-time SNMP data about a selected object. See "Configure SNMP Line Graph Actions" on page 1437.

Note: If you are displaying graphs for NNMi objects, the node or interface for which you want to graph information must support SNMPv1, SNMPv2c, or SNMPv3.

Use the showLineGraph syntax in a URL when you want to do any of the following:

- Display a Line Graph in an application other than NNMi.
- Display a Line Graph outside of an application and add it to your Favorites browser list.

To launch a Line Graph with the showLineGraph syntax, use the following URL:

> **Note:** If you copy/paste this URL, remove the spaces that were added for line-ending purposes in the documentation.

```
http://<serverName>:<portNumber>/nnm/launch?cmd=showLineGraph
[parameter list]
```

```
http://<serverName>:<portNumber>/nnm/launch?cmd=showLineGraph
&init=<x>&objtype=<x> &maxlines=<x> &maxtimerange=<x> &defaultsecs=
<x>&faststart=<true/false>
```

> **Note:** If the NNMi Web server uses the https protocol, use `https` instead of `http`. See the "Working with Certificates for NNMi" chapter in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.

<*serverName*> = the fully-qualified domain name of the NNMi management server (values allowed here are determined by the *Enable URL Redirect* setting in User Interface Configuration, see "Configuring the NNMi User Interface" on page 467)

<*portNumber*> = the NNMi HTTP port number

For a quick-reference list of all URL choices for launching NNMi, see **Help → Documentation Library → Integrate NNMi Elsewhere with URLs**. The Integrate NNMi Elsewhere with URLs page also contains a link to a list of sample URLs that you can copy/paste as a starting point.

Line Graph Parameters

Parameter	Description
`&init`	Use to define the lines you want displayed in the graph: • `instancelist` - Use to specify which instances of the SNMP MIB object to display. • One of the following for each line: ▪ `oid` - Use to specify the SNMP MIB object identifier value of each instance. ▪ `expr` - Use to specify the name of a MIB Expression that will be used for gathering the values on the Line Graph. • `label` - Use to specify the label to be used in the legend that describes each line on the graph.
`&objtype`	Use to specify the Object Type. For a Node Object Type, this value must be `${snmpAgent.id}`. For an Interface Object Type, this value must be `${hostedOn.snmpAgent.id}` **Note:** If you want to provide a Line Graph for a specified node, use the ID value for the node's SNMP Agent. NNMi displays the ID attribute value on the SNMP Agent form's Registration tab.
`&maxlines`	Use to specify the number of lines that NNMi should initially display on the Line Graph. To use the default value specified in the User Interface Configuration, omit this parameter.
`&maxtimerange`	Use to specify the number of hours for the Maximum Time Range in which the data in the Line Graph should be retained. After the Maximum Time Range number is reached, NNMi discards the oldest data point sets so that it can display the most recent data for the time range specified.
`&defaultsecs`	Use to specify the Polling Interval in which the graph data should be collected. To use the default value specified in the User Interface Configuration, omit this parameter.

Line Graph Parameters, continued

Parameter	Description
`&faststart`	Use to specify whether to increase the initial Polling Interval so that the initial data appears more quickly on the graph. Possible values are `true` or `false`. When you specify `true` for this option, NNMi increases the initial Polling Interval and then gradually decreases the Polling Interval until it reaches the Polling Interval configured for the graph. When you specify `false`, NNMi uses the Polling Interval set for the graph.
`&defaultfixedvertical`	Used to specify whether to lock the Y-axis. Possible values are `true` or `false`. When you specify `true`, the Y-axis remains fixed at the minimum and maximum values for the current set of data regardless of the time segment selected. This means NNMi does not automatically re-adjust the Y-axis to match the data values for the selected time segment. When you specify `false`, NNMi automatically adjusts the Y-axis to match the data values for the selected time segment.
`&ylabel`	Use to specify the label to be used for the Y-axis of the Line Graph.
more...	HP Network Node Manager i Software Smart Plug-ins (iSPIs) might provide more attributes to customize the line graph. See the documentation for the NNM iSPIs installed in your network environment.

Launch the Actions: Monitoring Settings Command

This URL is equivalent to the **Actions** → **Monitoring Settings** command in the console.

Launch the real-time results of the Monitoring configuration report. You must specify the target object.

Note: If the NNMi Web server uses the https protocol, use `https` instead of `http`. See the "Working with Certificates for NNMi" chapter in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.

`<serverName>` = the fully-qualified domain name of the NNMi management server (values allowed here are determined by the *Enable URL Redirect* setting in User Interface Configuration, see "Configuring the NNMi User Interface" on page 467)

`<portNumber>` = the NNMi HTTP port number

For a quick-reference list of all URL choices for launching NNMi, see **Help → Documentation Library → Integrate NNMi Elsewhere with URLs**. The Integrate NNMi Elsewhere with URLs page also contains a link to a list of sample URLs that you can copy/paste as a starting point.

(*NNMi Advanced*) If the Global Network Management feature is enabled and you are signed into a Global Manager:

- Node managed by the Global Manager = **Actions → Configuration Details → Monitoring Settings** opens a report, provided by the Global Manager (NNMi management server).
- Node managed by a Regional Manager = **Actions → Configuration Details → Monitoring Settings** accesses that Regional Manager (NNMi management server) and requests the report.

> **Note:** You must sign into that Regional Manager unless your network environment enables Single Sign-On (SSO) to that Regional Manager through the Global Manager. For more information, see the “Configuring Single Sign-On for Global Network Management” section in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.

To launch a window that displays a current Monitoring Settings report about a Node (SNMP Agent), use the following URL:

```
http://<
serverName
>:<
portNumber
>/nnm/launch?cmd=
runTool&tool=monitoringconf&objtype=SnmpAgent&nodename=<x>
```

> **Note:** If you copy/paste this URL, remove the spaces that were added for line-ending purposes in the documentation.

Monitoring Configuration Command Node Report Attributes

Attribute	Values
nodename	The node's DNS hostname (full or short) or IP address. If you use this attribute, NNMi tries to match the string you provide by following this procedure: • Check the value of the Hostname (*case-sensitive*) on the Node form. • Check the values in the Address column of the table on the Node form, Addresses tab, • Check the value of the System Name field on the in the Node form, General tab. • Check the value in the Name field on the Node form.

To launch a window that displays a current Monitoring configuration report about an Interface, use one of the following URLs:

NNMi displays the report for the first matching Interface found. Provide one or more attributes to ensure a unique match. See "Launch an Interface Form" on page 1542 for more information about each available attribute.

```
http://<serverName>:<portNumber>/nnm/launch?cmd=
runTool&tool=monitoringconf&objtype=Interface&objattrs=hostedOn.hostna
me=<x>;name=<x>
```

```
http://<serverName>:<portNumber
>/nnm/launch?cmd=runTool&tool=monitoringconf&objtype=Interface&objattr
s=hostedOn.hostname=<x>;ifName=<x>
```

```
http://<serverName>:<portNumber
>/nnm/launch?cmd=runTool&tool=monitoringconf&objtype=Interface&objattr
s=hostedOn.hostname=<x>;ifAlias=<x>
```

```
http://<serverName>:<portNumber
>/nnm/launch?cmd=runTool&tool=monitoringconf&objtype=Interface&objattr
s=hostedOn.hostname=<x>;ifIndex=<x>
```

Note: If you copy/paste this URL, remove the spaces that were added for line-ending purposes in the documentation.

Interface Form Attributes

Attribute	Values
hostedOn.hostname	The Hostname of the Node in which the interface resides. This is the Hostname attribute value from the associated Node form.
name	The Name attribute value from the Interface form.
ifName	The ifName attribute value from the Interface form.
ifAlias	The ifAlias attribute value from the Interface form.
ifIndex	The ifIndex attribute value from the Interface form.

To launch a window that displays a current Monitoring Settings report about an IP Address, use the following URL:

```
http://<
serverName
>:<
portNumber
>/nnm/launch?cmd=
runTool&tool=monitoringconf&objtype=IPAddress&objattrs=value=<x>
```

Note: If you copy/paste this URL, remove the spaces that were added for line-ending purposes in the documentation.

IP Address Form Attributes

Attribute	Values
value	The Address attribute value from the IP Address form.

To launch a window that displays a current Monitoring Settings report about an Card, use the following URL:

```
http://<
serverName
>:<
portNumber
>/nnm/launch?cmd=
runTool&tool=monitoringconf&objtype=Card&objattrs=value=<x>
```

Note: If you copy/paste this URL, remove the spaces that were added for line-ending purposes in the documentation.

Card Form Attributes

Attribute	Values
value	The card attribute value from the Card form.

To launch a window that displays a current Monitoring Settings report about a Router Redundancy Member (Instance), use the following URL:

```
http://<serverName>:<portNumber>/nnm/launch?
cmd=runTool&tool=monitoringconf&objtype=RouterRedundancyInstance&objid
=<x>
```

Note: If you copy/paste this URL, remove the spaces that were added for line-ending purposes in the documentation.

Monitoring Configuration Command Router Redundancy Member Report Attributes

Attribute	Values
objid	The Unique Object Identifier (unique per object type in the NNMi database). This attribute value is not visible on any form within the console. To find this value, use the nnmconfigexport.ovpl command-line tool to export an XML file that includes the `id` attribute value for each object instance.
objuuid	The Universally Unique Object Identifier (unique across all databases). This attribute value is not visible on any form within the console. To find this value, use the nnmconfigexport.ovpl command-line tool to export an XML file that includes the `uuid` attribute value for each object instance.

To launch a window that displays a current Monitoring Settings report about a Tracked Object, use the following URL:

```
http://<
serverName
>:<
portNumber
>/nnm/launch?
cmd=runTool&tool=monitoringconf&objtype=TrackedObject&objid=<x>
```

Note: If you copy/paste this URL, remove the spaces that were added for line-ending purposes in the documentation.

Monitoring Configuration Command Tracked Object Report Attributes

Attribute	Values
objid	The Unique Object Identifier (unique per object type in the NNMi database). This attribute value is not visible on any form within the console. To find this value, use the nnmconfigexport.ovpl command-line tool to export an XML file that includes the `id` attribute value for each object instance.
objuuid	The Universally Unique Object Identifier (unique across all databases). This attribute value is not visible on any form within the console. To find this value, use the nnmconfigexport.ovpl command-line tool to export an XML file that includes the `uuid` attribute value for each object instance.

To launch a window that displays a current Monitoring Settings report about a Node Component, use the following URL:

```
http://<
serverName
>:<
portNumber
>/nnm/launch?
cmd=runTool&tool=monitoringconf&objtype=ComponentHealth&objid=<x>
```

Note: If you copy/paste this URL, remove the spaces that were added for line-ending purposes in the documentation.

Monitoring Configuration Command Node Component Report Attributes

Attribute	Values
objid	The Unique Object Identifier (unique per object type in the NNMi database). NNMi displays the `id` attribute value on the Node form's Registration tab.
objuuid	The Universally Unique Object Identifier (unique across all databases). NNMi displays the `uuid` attribute value on the Node form's Registration tab.

Related Topics:

"Verify the Monitoring Settings" on page 416

Launch the Actions: Ping Command

This URL is equivalent to the **Actions** → **Ping (from server)** command in the console.

To launch a window that requests you to enter a node name, use the following URL:

`http://<serverName>:<portNumber>/nnm/launch?cmd=runTool&tool=ping`

> **Note:** If the NNMi Web server uses the https protocol, use `https` instead of `http`. See the "Working with Certificates for NNMi" chapter in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.

<serverName> = the fully-qualified domain name of the NNMi management server (values allowed here are determined by the *Enable URL Redirect* setting in User Interface Configuration, see "Configuring the NNMi User Interface" on page 467)

<portNumber> = the NNMi HTTP port number

For a quick-reference list of all URL choices for launching NNMi, see **Help** → **Documentation Library** → **Integrate NNMi Elsewhere with URLs**. The Integrate NNMi Elsewhere with URLs page also contains a link to a list of sample URLs that you can copy/paste as a starting point.

After you specify a node, the real-time results of the ping command appear.

To launch the real-time results of the ping command, use the following URL:

```
http://<
serverName
>:<
portNumber
>/nnm/launch?cmd=
runTool&tool=ping&timeoutSecs=<x>&numPings=<x>&nodename=<x>
```

(*NNMi Advanced*) If the Global Network Management feature is enabled and you are signed into a Global Manager:

- Node managed by the Global Manager = Actions → Ping issues an ICMP request from the Global Manager (NNMi management server).
- Node managed by a Regional Manager = Actions → Ping accesses that Regional Manager (NNMi management server) and issues the ICMP request.

> **Note:** You must sign into that Regional Manager unless your network environment enables Single Sign-On (SSO) to that Regional Manager through the Global Manager. For more information, see the “Configuring Single Sign-On for Global Network Management” section in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.

Ping Command Attributes

Attribute	Values
nodename	A DNS-resolvable hostname or IP address. The nodename value is passed literally to the underlying command you specify. The nodename value (in this case) is not required to correlate with anything in the NNM database.
timeoutSecs	Amount of time NNMi waits before abandoning a ping request.
numPings	Maximum number of retries.

Related Topics:

Test Node Access (Ping)

Launch the Actions: Status Details Command (for Node Groups)

This URL is equivalent to the **Actions** → **Status Details** command in the console.

For a quick-reference list of all URL choices for launching NNMi, see **Help** → **Documentation Library** → **Integrate NNMi Elsewhere with URLs**. The Integrate NNMi Elsewhere with URLs page also contains a link to a list of sample URLs that you can copy/paste as a starting point.

To launch a real-time calculation of current status for a specified Node Group, use the following URL:

```
http://<
serverName>:<portNumber>/nnm/launch?cmd=runTool&tool=nodegroupstatus
```

> **Note:** If the NNMi Web server uses the https protocol, use `https` instead of `http`. See the "Working with Certificates for NNMi" chapter in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.

<serverName> = the fully-qualified domain name of the NNMi management server (values allowed here are determined by the *Enable URL Redirect* setting in User Interface Configuration, see "Configuring the NNMi User Interface" on page 467)

<portNumber> = the NNMi HTTP port number

After you specify a node group, the real-time results of the node group's status calculation appear.

To launch a real-time calculation of current status for a specified Node Group and display a report of the information gathered, use the following URL:

```
http://<
serverName
>:<
portNumber>/nnm/launch?cmd=runTool&tool=nodegroupstatus&nodegroup=<x>
```

Filter by Node Group (launched Incident, Node, Interface, and IP Address views)

Attribute	Values
nodegroup	The *case-sensitive* Name attribute value of the Node Group to use as a filter for this view. **Note:** The Node Group name is translated. If your team shares NNMi within multiple locales, use nodegroupid or nodegroupuuid. If the Name value includes space characters, consider replacing the space character in your URL statement (see "W3C Rules for URLs" on page 1490).
nodegroupid	The `id` is the Unique Object Identifier (unique across the entire NNMi database). Provide the `id` of the Node Group to use as a filter for this view. This attribute value is not visible on any form within the console. To find this value, use the nnmconfigexport.ovpl command-line tool to export an XML file that includes the `id` attribute value for each object instance.
nodegroupuuid	The `uuid` is the Universally Unique Object Identifier (unique across all databases). Provide the `uuid` of the Node Group to use as a filter for this view. This attribute value is not visible on any form within the console. To find this value, use the nnmconfigexport.ovpl command-line tool to export an XML file that includes the `uuid` attribute value for each object instance.

Related Topics:

Check Status Details for a Node Group

Launch the Actions: Status Poll Command

This URL is equivalent to the **Actions** → **Polling** → **Status Poll** command in the console.

NNMi calculates the status of devices each time additional information is gathered. You can instruct NNMi to gather real-time data for all the information that NNMi uses to calculate Status for the specified Node. A window displays with a report about which information was gathered. The NNMi administrator determines the list of information gathered by establishing Monitoring configuration settings. See "Monitoring Network Health" on page 340 for more information.

> **Note:** To see the resulting Node status, see Verify Current Status of a Device.

For a quick-reference list of all URL choices for launching NNMi, see **Help** → **Documentation Library** → **Integrate NNMi Elsewhere with URLs**. The Integrate NNMi Elsewhere with URLs page also contains a link to a list of sample URLs that you can copy/paste as a starting point.

To launch a window that reports the current status for a node, use the following URL:

```
http://<
serverName>:<portNumber>/nnm/launch?cmd=runTool&tool=statuspoll
```

> **Note:** If the NNMi Web server uses the https protocol, use `https` instead of `http`. See the "Working with Certificates for NNMi" chapter in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.

`<serverName>` = the fully-qualified domain name of the NNMi management server (values allowed here are determined by the *Enable URL Redirect* setting in User Interface Configuration, see "Configuring the NNMi User Interface" on page 467)

`<portNumber>` = the NNMi HTTP port number

After you specify a node, the real-time results of the node's status appear.

To launch the real-time results of a node's status, use the following URL:

```
http://<
serverName
>:<portNumber>/nnm/launch?cmd=runTool&tool=statuspoll&nodename=<x>
```

Status Poll Command Attributes

Attribute	Values
nodename	The node's DNS hostname (full or short) or IP address. If you use this attribute, NNMi tries to match the string you provide by following this procedure: • Check the value of the Hostname (*case-sensitive*) on the Node form. • Check the values in the Address column of the table on the Node form, Addresses tab, • Check the value of the System Name field on the in the Node form, General tab. • Check the value in the Name field on the Node form.

Related Topics:

Verify Current Status of a Device

Launch the Actions: Trace Route Command

This URL is equivalent to the **Actions** → **Trace Route (from server)** command in the console.

To launch a window that requests you to enter a node name, use the following URL:

```
http://<
serverName>:<portNumber>/nnm/launch?cmd=runTool&tool=traceroute
```

> **Note:** If the NNMi Web server uses the https protocol, use `https` instead of `http`. See the "Working with Certificates for NNMi" chapter in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.

`<serverName>` = the fully-qualified domain name of the NNMi management server (values allowed here are determined by the *Enable URL Redirect* setting in User Interface Configuration, see "Configuring the NNMi User Interface" on page 467)

`<portNumber>` = the NNMi HTTP port number

For a quick-reference list of all URL choices for launching NNMi, see **Help → Documentation Library → Integrate NNMi Elsewhere with URLs**. The Integrate NNMi Elsewhere with URLs page also contains a link to a list of sample URLs that you can copy/paste as a starting point.

After you specify a node, the real-time results of the trace route command appear.

To launch the real-time results of the trace route command, use the following URL:

```
http://<
serverName
>:<portNumber>/nnm/launch?cmd=runTool&tool=traceroute&nodename=<x>
```

Trace Route Command Attributes

Attribute	Values
nodename	A DNS-resolvable hostname or IP address. The nodename value is passed literally to the underlying command you specify. The nodename value (in this case) is not required to correlate with anything in the NNM database.

Related topics:

Find the Route (traceroute)

Actions: Execute a Launch Action

The showMenuItem command launches a Menu Item that has been configured as a Launch Action in NNMi.

To execute a Launch Action requesting something from NNMi:

```
http://<serverName>:<portNumber>/nnm/launch?cmd=showMenuItem&key=<Menu
ItemKey>[&nodename=<hostname or IP_address>
```

> **Note:** If the NNMi Web server uses the https protocol, use `https` instead of `http`. See the "Working with Certificates for NNMi" chapter in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals.`

`<serverName>` = the fully-qualified domain name of the NNMi management server (values allowed here are determined by the *Enable URL Redirect* setting in User Interface Configuration, see "Configuring the NNMi User Interface" on page 467)

`<portNumber>` = the NNMi HTTP port number

For a quick-reference list of all URL choices for launching NNMi, see **Help → Documentation Library → Integrate NNMi Elsewhere with URLs**. The Integrate NNMi Elsewhere with URLs page also contains a link to a list of sample URLs that you can copy/paste as a starting point.

To execute a Launch Action requesting a script, application, or tool from your environment (not NNMi):

```
http://<
serverName>:<portNumber>/<application>?<yourURLparameter1>=${<attribut
e>}&<yourURLparameter2>=${<attribute>}
```

> **Note:** To extend the NNMi environment with additional applications, you must deploy them into a separate web-server or application-server on the same or different physical server from where the NNMi web-server or application-server is installed. See the *NNMi Developer's Toolkit* for more information.

`<serverName>` = the appropriate fully-qualified domain name

`<portNumber>` = the appropriate port number

> **Tip:** After you specify a *case-sensitive* Hostname, the real-time results of the Launch Action appear.

Trace Route Command Attributes

Attribute	Values
MenuItemKey	The Unique Key used for the Menu Item configuration. See "Configure Menu Item Basic Details" on page 1417 for more information.
nodename	*Optional*. A DNS-resolvable hostname or IP address indicating the node on which the action should be executed.

See "Configure Launch Actions" on page 1422 for information about creating a Launch Action.

Launch the Tools: MIB Browser (showMibBrowser)

Use the showMibBrowser URL to display the MIB Browser window and the SNMP getNext responses from one Node.

You must provide the Node name/IP-address and one MIB variable OID (Object Identifier) value to determine the starting point. This starting point can be any OID from any MIB file that is currently loaded onto the NNMi management server and supported by the SNMP agent for the specified node.

NNMi automatically gathers responses to all MIB objects from the designated OID down though the Internet MIB tree.

NNMi enables you to launch the MIB Browser in any of the following ways:

- Use the showMibBrowser syntax in a URL as described in this help topic.
- Click **Tools** → **MIB Browser** ("Determine the MIB Variables Supported for a Node (for Administrators)" on page 1454

- Click **Actions** → **MIB Information** →**Browse MIB** ("Determine the MIB Variables Supported for a Node (for Administrators)" on page 1454

To launch the MIB Browser, use the following URL:

```
http://<
serverName
>:<
portNumber
>/nnm/launch?cmd=showMibBrowser&node=<name|address>&oid=<name|number>
```

> **Note:** If the NNMi Web server uses the https protocol, use `https` instead of `http`. See the "Working with Certificates for NNMi" chapter in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.

<*serverName*> = the fully-qualified domain name of the NNMi management server (values allowed here are determined by the *Enable URL Redirect* setting in User Interface Configuration, see "Configuring the NNMi User Interface" on page 467)

<*portNumber*> = the NNMi HTTP port number

For a quick-reference list of all URL choices for launching NNMi, see **Help** → **Documentation Library** → **Integrate NNMi Elsewhere with URLs**. The Integrate NNMi Elsewhere with URLs page also contains a link to a list of sample URLs that you can copy/paste as a starting point.

MIB Browser Parameters

Parameter	Description
node	The node's DNS hostname (full or short) or IP address. If you use this attribute, NNMi tries to match the string you provide by following this procedure: • Check the value of the Hostname (*case-sensitive*) on the Node form. • Check the values in the Address column of the table on the Addresses tab in the Node form, Addresses tab, • Check the value of the System Name field on the in the Node form, General tab. • Check the value in the Name field on the Node form.
oid	Enter one of the following as defined in the associated MIB file. NNMi uses this OID as the starting displayed value and the starting point for the SNMP Walk feature in the SNMP MIB Browser window: • Textual name of the object, for example `.iso.org.dod.internet.mgmt.mib-2.ifMIB.ifMIBObjects.ifTestTable.ifTestEntry.ifTestResult` or `.1.3.6.1.2.1.31.1.3.1.ifTestResult`. • Numeric representation of the object as defined in the MIB file, for example `.1.3.6.1.2.1.31.1.3.1.4` (the numeric value must always begin with a period character).

MIB Browser Parameters, continued

Parameter	Description
node	The node's DNS hostname (full or short) or IP address. If you use this attribute, NNMi tries to match the string you provide by following this procedure: • Check the value of the Hostname (*case-sensitive*) on the Node form. • Check the values in the Address column of the table on the Addresses tab in the Node form, Addresses tab, • Check the value of the System Name field on the in the Node form, General tab. • Check the value in the Name field on the Node form.
	Tip: You can view the list of MIB variables provided by each available MIB file using the **Loaded MIBs** view. See "Loaded MIBs View" on page 1451 for more information.
Community String	*Optional*. In the **Community String** attribute, do one of the following: • Leave this attribute value blank. NNMi uses the Communication parameters currently configured in the NNMi database for the specified Node (if any). • Enter a valid *read community string* for the Node. NNMi uses default SNMP version, timeout, maximum retries, and port parameters provided by the NNMi administrator within the `nms-ui.properties` file. For more information, see the "Maintaining NNMi" chapter in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.
Walk	After the MIB Browser window displays, click the Walk button to issue an SNMP getNext for all MIB objects from the designated OID down though the Internet MIB tree.

Launch the Tools: NNMi Status Command

This URL is equivalent to the **Tools** → **NNMi Status** command in the console.

To launch a report of the current status of all NNMi processes and services, use the following URL:

```
http://<serverName>:<portNumber>/nnm/launch?cmd=runTool&tool=nnmstatus
```

> **Note:** If the NNMi Web server uses the https protocol, use `https` instead of `http`. See the "Working with Certificates for NNMi" chapter in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.

`<serverName>` = the fully-qualified domain name of the NNMi management server (values allowed here are determined by the *Enable URL Redirect* setting in User Interface Configuration, see "Configuring the NNMi User Interface" on page 467)

`<portNumber>` = the NNMi HTTP port number

For a quick-reference list of all URL choices for launching NNMi, see **Help → Documentation Library → Integrate NNMi Elsewhere with URLs**. The Integrate NNMi Elsewhere with URLs page also contains a link to a list of sample URLs that you can copy/paste as a starting point.

Related Topics:

"Verify that NNMi Processes Are Running" on page 81

Check the Status of NNMi

"NNMi Processes and Services" on page 81

Launch the File: Sign-Out Command

This URL is equivalent to the **File → Sign Out** command in the console.

To provide a link that issues a sign-out command, use the following URL:

```
http://<serverName>:<portNumber>/nnm/launch?cmd=signOut
```

> **Note:** If the NNMi Web server uses the https protocol, use `https` instead of `http`. See the "Working with Certificates for NNMi" chapter in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.

`<serverName>` = the fully-qualified domain name of the NNMi management server (values allowed here are determined by the *Enable URL Redirect* setting in User Interface Configuration, see "Configuring the NNMi User Interface" on page 467)

`<portNumber>` = the NNMi HTTP port number

This closes the user session and frees up any memory associated with the session.

Related Topics:

"Sign Out from the Console" on page 581

Launch the Tools: Sign-In/Out Audit Log Command

This URL is equivalent to the **Tools → Sign In/Out Audit Log** command in the console.

To launch a window that reports the current configuration for a node, use the following URL:

```
http://<
serverName>:<portNumber>/nnm/launch?cmd=runTool&tool=signinaudit
```

> **Note:** If the NNMi Web server uses the https protocol, use `https` instead of `http`. See the "Working with Certificates for NNMi" chapter in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.

<serverName> = the fully-qualified domain name of the NNMi management server (values allowed here are determined by the *Enable URL Redirect* setting in User Interface Configuration, see "Configuring the NNMi User Interface" on page 467)

<portNumber> = the NNMi HTTP port number

NNMi logs the history of sign-in and sign-out activity for each user since the NNMi management server was last restarted.

Related Topics:

"Audit NNMi User Activity" on page 586

Confirm that NNMi Is Running (cmd=isRunning)

To launch a message reporting whether NNMi is currently running, use the following URL:

```
http:/<serverName>:<portNumber>/nnm/launch?cmd=isRunning
```

> **Note:** If the NNMi Web server uses the https protocol, use `https` instead of `http`. See the "Working with Certificates for NNMi" chapter in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`.

<serverName> = the fully-qualified domain name of the NNMi management server (values allowed here are determined by the *Enable URL Redirect* setting in User Interface Configuration, see "Configuring the NNMi User Interface" on page 467)

<portNumber> = the NNMi HTTP port number

One of the following messages appears:

- NNMi is running.
- A browser error message that the URL is unreachable.

Chapter 19

Maintaining NNMi

As an NNMi administrator, you will want to perform the following tasks when maintaining NNMi configurations and data.

"Track Your NNMi Licenses" on the next page

"Extend a Licensed Capacity" on page 1575

"Export and Import Configuration Settings" on page 1579

"Back Up and Restore NNMi" on page 1596

"Archive and Delete Incidents" on page 1598

Check NNMi Health

As an NNMi administrator, you can check the status and overall health of NNMi using any of the following:

- Use **Help** → **System Information** to view NNMi and component health including NNMi's overall health status, information, and any issues related to the following:
  - Disk usage
  - Global Network Management (NNMi Advanced)
  - Memory
  - NNMi database
  - SNMP requests and queues
  - System resources

  > **Tip:** When HP Network Node Manager iSPI Performance for Metrics Softwareis installed, you can check Path View health. Display a Path View map (**Troubleshooting** → **Path View**). Select the starting and ending node for which you want to view the health information. Click **Actions** → **HP NNM iSPI Performance** → **Reporting - Path Health**. The nodes you select must reside in the NNMi topology database and be configured for performance measurement collection. See the NNM iSPI Performance for Metrics online help for more information.

  Click here for more information about NNMi's overall health status.

  NNMi uses the following statuses when monitoring its health:

NNMi Overall Health Status

Status	Description
Warning	Indicates performance issues that are not significantly affecting NNMi.
Minor	Indicates problems that might result in out of date data. For example, a component, such as State Poller might be out of synch because it is operating outside of expected ranges.
Major	Indicates problems that are significantly affecting the NNMi management server's operations, but are not yet critical. Major Status usually indicates that some action is required. For example, a trap threshold is reached.
Critcal	Indicates NNMi is not functioning. For example, NNMi is out of memory, all database connections are lost, or a major component has failed.

See Displaying NNMi System Information for more information.

- Use the **Tools** → **NNMi Self-Monitoring Graphs** to view information about NNMi components and their usage. NNMi Self-Monitoring Graphs include:
  - SNMP Trap Pipeline Rate
  - SNMP Trap Forwarding Rate
  - Discovery Progress
  - SNMP Requests

  Tip: Use the SNMP Requests Graph to tune Communication Configuration settings.

- NNMi administrators can use the command line on any NNMi management server to generate a report about NNMi health. See the nnmhealth.ovpl Reference Page for more information.

Track Your NNMi Licenses

To assist you in tracking your NNMi licenses, NNMi displays a status message at the bottom of the main console whenever the number of nodes in the database reaches your licensed capacity limit (compared to the number of nodes discovered). Install additional licenses (for 50 node increments or more) to extend the limit.

To see a report of the current number of discovered nodes and the current NNMi licensed capacity limit, access **View Licensing Information** from either of the following locations:

- **Help** → **About HP Network Node Manager i software**
- The Console Sign-In window

There are four categories of NNMi Software Licenses. Within each category, there are three types (instant-on, temporary, or permanent):

- Licenses for NNMi or NNMi Advanced:
  - Base (NNMi:Runtime)
  - iAdvanced

- Integration Enablement licenses. For example, required when connecting to HP Network Node Manager i Software Smart Plug-ins (iSPIs) on a remote server or extending the functionality of NNMi in other ways.
- Licenses for developers (SDK licenses).

When tracking license information, note the following:

- NNMi discovers and manages nodes up to the NNMi licensed capacity limit.
- If the number of discovered nodes reaches or exceeds the licensed capacity limit, NNMi randomly "Unmanages" nodes until the number of "Managed" nodes matches the licensed capacity limit. For example: this situation might occur when an Instant-On or Temporary license expires or when an incremental license is intentionally uninstalled from a particular server. No new nodes are discovered unless one of the following occurs:
  - Install a license extension, see "Extend a Licensed Capacity" below. (Any seeds that were "**Unmanaged**[1]" because of license issues, must be manually changed back to "Managed" after the license extension is installed.)
  - Review your configuration settings and limit NNMi discovery to only the important nodes in your network environment (see "Discovering Your Network" on page 175). Then, delete nodes and let NNMi rediscovery reset the managed inventory of nodes (see "Delete Nodes" on page 1602).
- NNMi generates Incidents under the following circumstances:
  - The number of discovered nodes exceeds the current licensed capacity limit.
  - An Instant-On or Temporary license expires.
  - HP Network Node Manager i Software Smart Plug-ins (iSPIs) are purchased and installed on the NNMi management server. However, the NNMi licensed capacity limit does not match the NNM iSPI licensed capacity limit. See "Purchase an HP Network Node Manager i Smart Plug-in" on page 1486 for more information about the NNM iSPIs.
  - The NNMi licensed capacity limit does not match the required Integration Enablement licensed capacity limit. For example, when connecting to HP Network Node Manager i Software Smart Plug-ins (iSPIs) on a remote server or when extending the functionality of NNMi in other ways by using the NNMi SDK.

Related Topics:

"Extend a Licensed Capacity" below

"Purchase an HP Network Node Manager i Smart Plug-in" on page 1486

"Integrations with HP and Third-Party Products" on page 1487

Extend a Licensed Capacity

To extend the licensed capacity, purchase and install an additional NNMi or NNMi Advanced Software License.

[1] Indicates the Management Mode is "Not Managed" or "Out of Service".

Contact your HP Sales Representative or your Authorized Hewlett-Packard Reseller for information about the NNMi licensing structure, and to learn how to add license tiers for enterprise installations. To obtain additional license keys, go to the HP License Key Delivery Service: https://webware.hp.com/welcome.asp

For more information, see the *HP Network Node Manager i Software Interactive Installation Guide* and nnmlicense.ovpl for more information.

> **Note:** The licensed capacity count is cumulative for each licensed product (across all installed license keys for that licensed product).

After you purchase a software license, install the NNMi Software License key using one of the following methods:

- **From the command line**:
  a. At the command prompt for the NNMi management server, type the following (see the nnmlicense.ovpl Reference Page for more information):

     For <*product*>, use one of the following: NNM, iSPI-NET, iSPI-Points, or PerfSPI

     - **Windows**:
       `%NnmInstallDir%\bin\nnmlicense.ovpl <product> -f <license_file>`
     - **UNIX**:
       `opt/OV/bin/nnmlicense.ovpl <product> -f <license_file>`

  b. NNMi automatically completes the installation.
- **Using Autopass and your HP Order Number (not possible behind a firewall)**:
  a. Open the Autopass user interface. At the command line for the NNMi management server, type the following (see the nnmlicense.ovpl Reference Page for more information):

     For <*product*>, use one of the following: NNM, iSPI-NET, iSPI-Points, or PerfSPI

     - **Windows**:
       `%NnmInstallDir%\bin\nnmlicense.ovpl <product> -gui`
     - **UNIX**:
       `opt/OV/bin/nnmlicense.ovpl <product> -gui`

  b. On the left side of the Autopass window, click **License Management**.

  c. Click **Install License Key**.

  d. Click **Retrieve/Install License Key**.

  e. Enter your HP Order Number and follow the Autopass prompts to complete the License key retrieval process.

  f. Autopass automatically completes the installation.

Related Topics:

"Track Your NNMi Licenses" on page 1574

"Purchase an HP Network Node Manager i Smart Plug-in" on page 1486

"Integrations with HP and Third-Party Products" on page 1487

"Delete Nodes" on page 1602

Resolve Inconsistencies between State and Status

At times, differences between State and Status values might occur. These differences generally indicate the system is busy processing large volumes of data, perhaps due to a significant configuration change or network outage.

Note the following:

- NNMi updates State before Status. A delay in Status updates might be due in part to the processing required for root cause analysis performed by the Causal Engine.
- As NNMi completes this processing, the consistency of State and Status is restored.
- Associated incidents might also be delayed during this time.
- Status updates can run behind by the amount of time listed for *Delay Processing Input* in the **System Information** dialog's **Causal Engine** tab. See Displaying NNMi System Information.

If the consistency of State and Status is not restored, NNMi enables you to correct the State or Status inconsistencies for each of the following:

Resolve State or Status Inconsistencies on a Single Node

To correct either an unexpected State or inconsistent Status value on a node:

1. Navigate to the node view or map of interest and right-click the node

2. Select **Polling** > **Configuration Poll**

3. Select **Polling** > **Status Poll**

To correct inconsistent Status values on a node for which the State value is correct:

1. Navigate to the nodes view or map of interest and right-click the node

2. Select **Polling** > **Status Poll**

Resolve State or Status Inconsistencies on Multiple Nodes

To correct State or Status inconsistencies on multiple nodes, use the nnmnoderediscover.ovpl command.

The nnmnoderediscover.ovpl command, when used with the –fullsync option, enables you to re-synchronize the nodes specified.

The re-synchronization process performs the following for each node:

- Rediscovers the node.
- Reloads and refreshes the monitoring configuration for the node.
- Reanalyzes State and Status for the node.

To re-synchronize a single node:

```
nnmnoderediscover.ovpl -node hostname -fullsync
```

To re-synchronize all nodes in a file:

```
nnmnoderediscover.ovpl -file filename -fullsync
```

Tip: To re-synchronize the nodes in a Node Group, first use the nnmnodegroup.ovpl command to populate a file with the names of the nodes for a specified Node Group. Then, use the file created as the *filename* argument to nnmnoderediscover.ovpl.

To re-syncrhonize all the nodes on an NNMi management server:

```
nnmnoderediscover.ovpl -all -fullsync
```

If a large number of nodes (for example, thousands) are being re-synchronized, it is recommended that you re-synchronize these nodes during off-peak periods, when possible.

NNMi must remain running until the re-sychronization is complete. If NNMi is stopped before the re-synchronization is complete, run the nnmnoderediscover.ovpl command again and allow the re-synchronization to complete.

For more information, see nnmnoderediscover.ovpl.

If you have a Global Network Management environment, also see "Node Synchronization Issues " on page 115

About Environment Variables

These are the default values for NNMi environment variables. Actual values depend on the selections made during NNMi installation. See the nnm.envvars Reference Page for more information.

Operating System	Environment Variable Values

Windows Server 2008	`%NnmInstallDir% =`
	`<drive>\Program Files(x86)\HP\HP BTO Software\`
	`%NnmDataDir% =`
	`<drive>\ProgramData\HP\HP BTO Software\`
	<drive> is the location where NNMi was installed.
	Note: On Windows systems, the NNMi installation process creates these environment variables so they are always available.

UNIX	`$NnmInstallDir = /opt/OV/`
	`$NnmDataDir = /var/opt/OV/`
	Note: On UNIX systems, you must manually create these environment variables if you want to use them. See the HP Network Node Manager i Software Deployment Reference, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals` and the nnm.envvars Reference Page for more information.

Export and Import Configuration Settings

See the nnmconfigexport.ovpl and nnmconfigimport.ovpl Reference Pages for more information, including the complete list of the command line arguments for each command.

The choices that you make when exporting NNMi configuration settings determine how that configuration information can be used. For example:

- Export a copy of the existing NNMi configuration settings before you try experimenting with a new idea. You can use that exported file to restore your configuration settings if your experiment does not work the way you thought it would work.
- Export the NNMi configuration settings from a server in your test environment. Import those configuration settings onto the NNMi management server that your team will use to manage your network environment.
- (*NNMi Advanced - Global Network Management feature*) Export configuration settings to share configuration settings among the Regional Managers in your network environment (for example, Node Group definitions, Trap Forward to Global Managers settings, and NNM 6.x/7.x Event Incidents Forward to Global Managers settings).

Carefully review the following topics to make an informed choice:

"Export/Import Behavior and Dependencies" below

"Export a Snapshot of Your Configuration Settings" on page 1584

"Import Configuration Files to Restore Previous Settings" on page 1586

"Transfer Configuration Settings to Another NNMi Management Server" on page 1588

"Troubleshooting Imports of Configuration Files" on page 1590

Export/Import Behavior and Dependencies

Your configuration settings can be exported to make a copy, and then imported onto the same NNMi management server or another NNMi management server. The locale setting on the NNMi management server at the time of Export must match the locale setting on the NNMi management server at the time of Import.

You need to understand the behavior and dependencies (see the table). The choices that you make when exporting NNMi configuration settings determine how that configuration information can be used:

Replaces all. Export files with this behavior make changes to the NNMi database when Imported (click here for more information).

- NNMi replaces all object instances with matching key identifiers (see "Troubleshooting Imports of Configuration Files" on page 1590 for information about key identifiers).
- NNMi adds all object instances with key identifiers that do not exist in the NNMi database
- ***NNMi deletes all existing object instances with key identifiers that do not match any in the exported file***.

Incremental. Export files with this behavior make changes to the NNMi database when Imported (click here for more information).

- NNMi updates all object instances with matching key identifiers (see "Troubleshooting Imports of Configuration Files" on page 1590 for information about key identifiers).

  Caution: NNMi also overwrites the values of any codes associated with these object instances (for example, incident family).

- NNMi adds all object instances with key identifiers that do not exist in the NNMi database.
- NNMi does not touch existing object instances with key identifiers that do not match any in the exported file.

Incremental (subset). Export files with this behavior include configuration changes that were made by one Author. Export files with this behavior make changes to the NNMi database when Imported (click here for more information).

- NNMi updates all object instances with matching key identifiers (see "Troubleshooting Imports of Configuration Files" on page 1590 for information about key identifiers).

  Caution: NNMi also overwrites the values of any codes associated with these object instances (for example, incident family).

- NNMi adds all object instances with key identifiers that do not exist in the NNMi database.
- NNMi does not touch existing object instances with key identifiers that do not match any in the exported file.

Export/Import Behavior and Dependencies Among Configuration Areas

Configuration Workspace's View Name	Export Option	Import Behavior	Dependencies
Author *	-c author	Incremental	No dependencies. Import requires one Export file (author.xml). * Not a workspace, but an important data object.
	-c author -a < *authorUniqueKey* >	Incremental (subset)	No dependencies. Import requires one Export file (author.xml).
Communication	-c comm	Replaces all	No dependencies. Import requires one Export file (comm.xml). **Caution**: SNMPv3 configuration settings cannot be exported because SNMPv3 data is encrypted based on the NNMi encryption key (generated during NNMi installation). Therefore, the SNMPv3 encrypted data cannot be imported into another installed version of NNMi because the

Export/Import Behavior and Dependencies Among Configuration Areas, continued

Configuration Workspace's View Name	Export Option	Import Behavior	Dependencies
			encryption key is different.
Custom Correlations	-c customCorrelation	Incremental	
	-c device -a < *authorUniqueKey* >	Incremental (subset)	The required Author information is embedded in the Export file.
Custom Poller	-c custpoll	Incremental	Import requires four Export files, and they must be imported in this order: (1) author.xml, (2) device.xml, (3) nodegroup.xml, and (4) custpoll.xml **Note**: When importing modifications to an existing Custom Poller Collection, NNMi sets the **Active State** for all associated Policies to **Suspended**.
Device Profiles	-c device	Incremental	Import requires two Export files, and they must be imported in this order: (1) author.xml and (2) device.xml
	-c device -a < *authorUniqueKey* >	Incremental (subset)	Import requires one Export file (device.xml). The required Author information is embedded in the Export file.
Discovery	-c disco	Replaces all	Import requires seven Export files, and they must be imported in this order: (1) comm.xml, (2) discoseed.xml, (3) iftype.xml, (4) author.xml, (5) device.xml, (6) ifgroup.xmland, and (7) disco.xml
Discovery Seeds	-c discoseed	Incremental	Import requires two Export files, and they must be imported in this order: (1) comm.xml and (2) discoseed.xml
Global Network Management			No export/import permitted at this time.
Icons and icon images	-c icons	Incremental	No dependencies. Import requires one Export file (icons.xml)

Export/Import Behavior and Dependencies Among Configuration Areas, continued

Configuration Workspace's View Name	Export Option	Import Behavior	Dependencies
	-c icons -a < *authorUniqueKey* >	Incremental (subset)	No dependencies. Import requires one Export file (icons.xml)
Incident	-c incident	Incremental	Import requires two Export files, and they must be imported in this order: (1) author.xml and (2) incident.xml
	-c incident -a < *authorUniqueKey* >	Incremental (subset)	Import requires one Export file (incident.xml). The required Author information is embedded in the Export file.
Interface Groups	-c ifgroup	Incremental	Import requires five Export files, and they must be imported in this order: (1) iftype.xml, (2) author.xml, (3) device.xml, (4) nodegroup.xml, and (5) ifgroup.xml
ifTypes	-c iftype	Incremental	Interface Types. No dependencies. Import requires one Export file (iftype.xml).
Management Stations (6.x/7.x)	-c station	Incremental	NNM 6.x or 7.x Management Stations. No dependencies. Import requires one Export file (station.xml).
Menus	-c menu	Incremental	Import requires two Export files, and they must be imported in this order: (1) author.xml and (2) menu.xml
	-c menu -a < *authorUniqueKey* >	Incremental (subset)	Import requires one Export file (menu.xml). The required Author information is embedded in the Export file.
Menu items (formally URL Actions)	-c menuitem (formally -c urlaction)	Incremental	Import requires two Export files, and they must be imported in this order: (1) author.xml and (2) menuitem.xml
	-c menuitem -a < *authorUniqueKey* >	Incremental (subset)	Import requires one Export file (menuitem.xml). The required Author information is embedded in the Export file.

Export/Import Behavior and Dependencies Among Configuration Areas, continued

Configuration Workspace's View Name	Export Option	Import Behavior	Dependencies
	(formally -c urlaction -a < *authorUniqueKey* >)		
MIB Expressions	-c mibexpr	Incremental	Import requires two Export files, and they must be imported in this order: (1) author.xml and (2) mibexpr.xml
	-c mibexpr -a < *authorUniqueKey* >	Incremental (subset)	Import requires one Export file (mibexpr.xml). The required Author information is embedded in the Export file.
MIB OID Types	-c mibtypes	Incremental	
Monitoring	-c monitoring	Replaces all	Import requires six Export files, and they must be imported in this order: (1) author.xml, (2) device.xml, (3) nodegroup.xml, (4) iftype.xml, (5) ifgroup.xml, and (6) monitoring.xml
Node Groups	-c nodegroup	Incremental	Import requires three Export files, and they must be imported in this order: (1) author.xml, (2) device.xml, and (3) nodegroup.xml **Caution**: Island Node Groups are never exported. See "Island Node Groups" on page 337.
Node Group Map Settings	-c ngmap	Incremental	Import requires four Export files, and they must be imported in this order: (1) author.xml, (2) device.xml, (3) nodegroup.xml, and (4) ngmap.xml **Note**: Any time you save a map layout, NNMi deletes any previous node locations. Therefore, each export contains only the node locations that were last saved.
RAMS Servers	-c rams	Incremental	HP Router Analytics Management Systems data from the RAMS Servers

Export/Import Behavior and Dependencies Among Configuration Areas, continued

Configuration Workspace's View Name	Export Option	Import Behavior	Dependencies
			view (does not include data from the Integration Module Configuration *HP RAMS MPLS WAN*). No dependencies. Import requires one Export file (rams.xml).
Security Groups Tenants	-c security		Exports Security Groups and Tenants.
Security Group Mappings	-c securitymappings		Exports Security Group Mappings.
Status	-c status	Replaces all	No dependencies. Import requires one Export file (status.xml). The imported status applies to all Node Groups in the database.
Traps	-c trap	Incremental	Import requires three Export files: (1) author.xml, (2) incident.xml, and (3) nodegroup.xml
	-c trap -a < *authorUniqueKey* >	Incremental (subset)	The required Author and Node Group information is embedded in the Export file.
User Accounts User Account Mappings User Groups NNMi Roles	-c account	Incremental	Exports User Accounts, NNMi Roles, User Groups, and User Account Mappings. This command gathers data from multiple Configuration workspace views. Import requires one Export file (account.xml). The data from all the Configuration workspace views is embedded in the Export file.
User Interface	-c ui	Incremental	No dependencies. Import requires one Export file (ui.xml).

Export a Snapshot of Your Configuration Settings

If you export your configuration settings before you begin making changes, you can easily "undo" your changes if you decide that you do not like the results.

To export a snapshot of your configuration settings:

1. Import behavior is determined when you generate the Export files. Make sure you understand your Import behavior choices before you begin. See "Export/Import Behavior and Dependencies" on page 1579 (consider printing that topic for reference).

   Caution: The locale setting on the NNMi management server at the time of Export must match the locale setting on the NNMi management server at the time of Import. Otherwise, you risk losing data. Also, make sure that both NNMi management servers have the same NNMi version/patch number.

2. A user name and password are required with the export command:

   If you do not want to enter an NNMi User Name attribute value and an NNMi Password attribute value at the command line, you can use the nnmsetcmduserpw.ovpl command to specify the valid user name and password (instead of `-u` and `-p`). The credentials set using the `nnmsetcmduserpw.ovpl` command are valid for command execution by the same user. See "Set Up Command Line Access to NNMi" on page 577 for more information.

   ```
   -u <NNMiadminUserName> -p <NNMiadminPassword>
   ```

3. Check whether the configuration settings you want to export have dependencies, see "Export/Import Behavior and Dependencies" on page 1579.

   - If no dependencies, export only the configuration settings you are planning to change.
   - If yes, decide whether you need a copy of the dependencies (only if you plan to make changes to those configuration settings, as well). Then export all the required files.

4. At the command line of the NNMi management server, type the command to generate the required export files.

   - To export all configuration settings, use the following command:

     ```
     nnmconfigexport.ovpl -c -all -f <directory>
     ```

     You can use `-x <file_prefix>` to provide a unique prefix to the set of exported files. For example, use today's date or a clue about the reason you needed the files:

     ```
     nnmconfigexport.ovpl -c -all -f <directory> -x <file_prefix>
     ```

   - To export specific configuration settings <*X*> from multiple configuration workspace views, separate each with a comma (see "Export/Import Behavior and Dependencies" on page 1579 or the nnmconfigexport.ovpl Reference Pages for the list of choices):

     ```
     nnmconfigexport.ovpl -c <X>, <X>, <X> -f <directory>
     ```

     You can use `-x <file_prefix>` to provide a unique prefix to the set of exported files. For example, use today's date or a clue about the reason you needed the files:

     ```
     nnmconfigexport.ovpl -c <X>, <X>, <X> -f <directory> -x <file_
     prefix>
     ```

   - To export configuration settings that were created by a particular author (for Author, Device Profiles, Incident, or URL Actions), add the `-a <authorUniqueKey>` attribute to the command and provide the Unique Key.

     Note: Only one author per `-a <authorUniqueKey>` export command is allowed.

Find the Unique Keys for all authors by exporting an author.xml file, then open the file in a text editor and locate the Key attribute values.

Find the Unique Key for a particular Author, in the NNMi console:

i. Open one of these Configuration workspaces in the NNMi console: Device Profiles Configuration, Incidents, or URL Actions.

ii. Select an object created by the Author of interest.

iii. Display the Author form, and copy the value of the Unique Key attribute.

5. Verify that the required xml files are in the specified directory.

   Caution: Do not edit the exported file before importing.

You are now ready to make configuration changes.

To undo your configuration setting changes, see "Import Configuration Files to Restore Previous Settings" below.

Import Configuration Files to Restore Previous Settings

If you have a set of export files, you can change the Configuration settings on your NNMi management server to match the settings in the exported files.

Note: You can change the names of the exported files before importing. The import still works the same.

Caution: Do not edit the exported file before importing.

To import a previous snapshot of your configuration settings:

1. Import behavior is determined when you generate the Export files. Make sure your exported files were generated in a manner that meets your current needs. See "Export/Import Behavior and Dependencies" on page 1579 (consider printing that topic for reference).

   Caution: The locale setting on the NNMi management server at the time of Export must match the locale setting on the NNMi management server at the time of Import. Otherwise, you risk losing data. Also, make sure that both NNMi management servers have the same NNMi version/patch number.

2. A user name and password are required with the import command:

   If you do not want to enter an NNMi User Name attribute value and an NNMi Password attribute value at the command line, you can use the nnmsetcmduserpw.ovpl command to specify the valid user name and password (instead of `-u` and `-p`). The credentials set using the `nnmsetcmduserpw.ovpl` command are valid for command execution by the same user. See "Set Up Command Line Access to NNMi" on page 577 for more information.

   ```
   -u <NNMiadminUserName> -p <NNMiadminPassword>
   ```

3. Check whether the configuration settings you want to import have dependencies, see "Export/Import Behavior and Dependencies" on page 1579.

- If no dependencies, import only the configuration settings you are planning to change.
- If yes, decide if you need a copy of the dependencies (only if you made changes to those configuration settings, as well). Then import all the required files.

4. At the command line of the NNMi management server, type the command to import a file:

   `nnmconfigimport.ovpl -f <filename>`

   When importing multiple XML files at once using `-f <directory>`, the NNMi `nnmconfigimport.ovpl` command takes care of ordering issues.

   - To import all configuration settings, use the following command:

     `nnmconfigimport.ovpl -f <directory>`

     You can use `-x <file_prefix>` to specify a unique prefix for a set of files:

     `nnmconfigimport.ovpl -f <directory> -x <file_prefix>`

   - To import specific configuration settings from multiple configuration areas, create a directory that contains the set of files you want to import.

     At the command line of the NNMi management server, type the appropriate command to import files:

     `nnmconfigimport.ovpl -f <directory>`

     You can use `-x <file_prefix>` to specify a unique prefix for a set of files:

     `nnmconfigimport.ovpl -f <directory> -x <filePrefix>`

   - To import configuration settings that were created by specific authors (for Author, Device Profiles, Incident, or URL Actions), create a directory that contains the set of files you want to import.

     At the command line of the NNMi management server, type the appropriate command to import the files:

     `nnmconfigimport.ovpl -f <file>`

     `nnmconfigimport.ovpl -f <directory>`

     You can use `-x <file_prefix>` to provide a unique prefix for a set of exported files:

     `nnmconfigimport.ovpl -f <directory> -x <filePrefix>`

5. If you encounter problems, see "Troubleshooting Imports of Configuration Files" on page 1590.

   Additional import options for timeout or memory issues:

   You can append the following options to any import command if you encounter problems:

Option	Description	Default Setting
`-timeout <seconds>`	For larger data imports, you might encounter timeout issues. To increase the number of seconds that NNMi waits (per file) during an import, append the `-timeout` option to the end of your command line.	1800 seconds (minimum)

Option	Description	Default Setting
`-memory <` *megabytes* `>`	For larger data imports, you might encounter memory issues. To increase the number of megabytes allotted to memory during an import, append the `-memory` option to the end of your command line.	512 megabytes

6. After completing the import, open NNMi and verify your configuration settings.

Transfer Configuration Settings to Another NNMi Management Server

You can export configuration settings and import them onto another NNMi management server to save time.

Note: You can change the names of the exported files before importing. The import still works the same.

Caution: Do not edit the exported file before importing.

To move configuration settings to another NNMi management server:

1. Import behavior is determined when you generate the Export files. Make sure you understand your Import behavior choices before you begin. See "Export/Import Behavior and Dependencies" on page 1579 (consider printing that topic for reference).

   Caution: The locale setting on the NNMi management server at the time of Export must match the locale setting on the NNMi management server at the time of Import. Otherwise, you risk losing data. Also, make sure that both NNMi management servers have the same NNMi version/patch number.

2. A user name and password are required with the export command:

   If you do not want to enter an NNMi User Name attribute value and an NNMi Password attribute value at the command line, you can use the nnmsetcmduserpw.ovpl command to specify the valid user name and password (instead of `-u` and `-p`). The credentials set using the `nnmsetcmduserpw.ovpl` command are valid for command execution by the same user. See "Set Up Command Line Access to NNMi" on page 577 for more information.

   ```
   -u <NNMiadminUserName> -p <NNMiadminPassword>
   ```

3. Check whether the configuration settings you want to export have dependencies, see "Export/Import Behavior and Dependencies" on page 1579.

   - If no dependencies, export only the configuration settings you are planning to change.
   - If yes, decide if you need a copy of the dependencies (only if you plan to make changes to those configuration settings, as well). Then export all the required files.

4. At the command line of the NNMi management server export all configuration settings, type the appropriate command:

```
nnmconfigexport.ovpl –c -all -f <directory>
```

You can use `-x <file_prefix>` to provide a unique prefix to the set of exported files. For example, use today's date or a clue about the reason you needed the files:

```
nnmconfigexport.ovpl –c -all -f <directory> -x <file_prefix>
```

Note: You can change the names of the exported files before importing. The import still works the same.

5. Delete any files that you do not need.
6. To export configuration settings that were created by a particular author (for Author, Device Profiles, Incident, or URL Actions), repeat the export command for each configuration item modified by that author. Add the `–a <authorUniqueKey>` attribute to the command and provide the Unique Key.

   Note: Only one author per `–a <authorUniqueKey>` export command is allowed.

   ```
   nnmconfigimport.ovpl -a <authorUniqueKey> -f <directory>
   ```

   You can use `-x <file_prefix>` to provide a unique prefix to the set of exported files. For example, use today's date or a clue about the reason you needed the files:

   ```
   nnmconfigimport.ovpl -a <authorUniqueKey> -f <directory> -x <file_
   prefix>
   ```

   Find the Unique Keys for all authors by exporting an author.xml file, then open the file in a text editor and locate the Key attribute values.

   Find the Unique Key for a particular Author, in the NNMi console:

   a. Open one of these Configuration workspaces in the NNMi console: Device Profiles Configuration, Incidents, or URL Actions.
   b. Select an object created by the Author of interest.
   c. Display the Author form, and copy the value of the Unique Key attribute.
7. Verify that all required xml files are in the specified directory.

To import the configuration settings onto the other NNMi management server:

1. Import behavior is determined when you generate the Export files. Make sure you understand your Import behavior choices before you begin. See "Export/Import Behavior and Dependencies" on page 1579 (consider printing that topic for reference).

   Caution: The locale setting on the NNMi management server at the time of Export must match the locale setting on the NNMi management server at the time of Import. Otherwise, you risk losing data. Also, make sure that both NNMi management servers have the same NNMi version/patch number.
2. A user name and password are required with the export command:

   If you do not want to enter an NNMi User Name attribute value and an NNMi Password attribute value at the command line, you can use the nnmsetcmduserpw.ovpl command to specify the valid user name and password (instead of `–u` and `–p`). The credentials set using the `nnmsetcmduserpw.ovpl` command are valid for command execution by the same user. See "Set Up Command Line Access to NNMi" on page 577 for more information.

```
-u <NNMiadminUserName> -p <NNMiadminPassword>
```

3. Verify that all required xml files are in the specified directory.
4. When importing multiple XML files at once using `-f <directory>`, the NNMi `nnmconfigimport.ovpl` command takes care of ordering issues.

   At the command line of the NNMi management server, type the appropriate command to import the configuration files that you gathered for transfer:

   ```
   nnmconfigimport.ovpl -f <filename>
   ```

   ```
   nnmconfigimport.ovpl -f <directory>
   ```

   You can use `-x <file_prefix>` to specify a unique prefix for a set of files:

   ```
   nnmconfigimport.ovpl -f <directory> -x <file_prefix>
   ```

5. If you encounter problems, see "Troubleshooting Imports of Configuration Files" below.

   Additional import options for timeout or memory issues:

   You can append the following options to any import command if you encounter problems:

Option	Description	Default Setting
`-timeout <seconds>`	For larger data imports, you might encounter timeout issues. To increase the number of seconds that NNMi waits (per file) during an import, append the `-timeout` option to the end of your command line.	1800 seconds (minimum)
`-memory <megabytes>`	For larger data imports, you might encounter memory issues. To increase the number of megabytes allotted to memory during an import, append the `-memory` option to the end of your command line.	512 megabytes

6. After completing the import, open NNMi and verify your configuration settings.

Troubleshooting Imports of Configuration Files

When importing incremental sets of configuration files, NNMi abandons the import if mismatched configuration objects are encountered. Each configuration object has a set of Unique Identifier values that must match for incremental updates, or must not match any existing data before NNMi adds the configuration object to the database, see tables below.

If you receive an error message while trying to import configuration information, use the information below to figure out how to use the error message to determine what to change before creating another export file (thus, solving the problem).

Note: You can change the names of the exported files before importing. The import still works the same.

Caution: Do not edit the exported file before importing.

Author Configuration Unique Identifiers

Configuration Item Name	Primary Identifier	Editable Attribute?	Other Key Attributes that must be unique
Author = Author form	author = Unique Key attribute value	Yes	

Communication Configuration Unique Identifiers

Attribute Name = NNMi Console Name	Primary Identifier	Editable Attribute?	Other Key Attributes that must be unique

CommunicationRegion = Communication Region form	uuid	No	name = Name value
			ordering = Ordering value
			Caution: SNMPv3 configuration settings cannot be exported because SNMPv3 data is encrypted based on the NNMi encryption key (generated during NNMi installation). Therefore, the SNMPv3 encrypted data cannot be imported into another installed version of NNMi because the encryption key is different.

Custom Correlation Configuration Unique Identifiers

Attribute Name	Primary Identifier	Editable Attribute?	Other Key Attributes that must be unique
CausalCorrelation = Causal Rule Form	uuid	No	
GeneralizedCorrelation = Correlation Rule Form	uuid	No	

Custom Poller Configuration Unique Identifiers

Attribute Name	Primary Identifier	Editable Attribute?	Other Key Attributes that must be unique
ComparisonMap = Comparison Map form	uuid	No	ordering = Ordering value
Policy = Custom Poller Policy form	uuid	No	The combination of these two:

Custom Poller Configuration Unique Identifiers, continued

Attribute Name	Primary Identifier	Editable Attribute?	Other Key Attributes that must be unique

			collection = Collection value
			ordering = Ordering value

Device Profile Configuration Unique Identifiers

Attribute Name	Primary Identifier	Editable Attribute?	Other Key Attributes that must be unique
DeviceCategory = Device Category form	key = Unique Key attribute	Yes	
DeviceFamily = Device Family form	key = Unique Key attribute	Yes	
DeviceProfile = Device Profile form	snmpObjectId = SNMP Object ID value	Yes	
DeviceVendor = Device Vendor form	key = Unique Key attribute	Yes	

Discovery Configuration Unique Identifiers

Attribute Name	Primary Identifier	Editable Attribute?	Other Key Attributes that must be unique

AutoDiscoveryRegion = Auto-Discovery Rule form	uuid	No	name = Name value
			ordering = Ordering value

Discovery Seed Configuration Unique Identifiers

Attribute Name	Primary Identifier	Editable Attribute?	Other Key Attributes that must be unique
DiscoverySeed = Discovery Seed form	host = Hostname (*not case-sensitive*) / IP Address value	Yes	

Incident Configuration Unique Identifiers

Attribute Name	Primary Identifier	Editable Attribute?	Other Key Attributes that must be unique
MgmtEventConfig = Management Event Configuration form	uuid	No	name = Name value

Incident Configuration Unique Identifiers, continued

Attribute Name	Primary Identifier	Editable Attribute?	Other Key Attributes that must be unique

SnmpTrapConfig = SNMP Trap Configuration form	uuid	No	iod = SNMP Object ID value
			name = Name value

RemoteNnmEventConfig	uuid	No	iod name = Name value

PairwiseConfig = Pairwise Configuration form	uuid	No	name = Name value
			The combination of these two:
			firstIncidentName = First Incident Configuration value
			secondIncidentName = Second Incident Configuration value

Interface Groups Configuration Unique Identifiers

Attribute Name	Primary Identifier	Editable Attribute?	Other Key Attributes that must be unique
InterfaceGroup = Interface Group form	uuid	No	name = Name value

Interface Type Configuration Unique Identifiers

Attribute Name	Primary Identifier	Editable Attribute?	Other Key Attributes that must be unique
ifType	ifType attribute	Yes	

Menus Unique Identifiers

Attribute Name	Primary Identifier	Editable Attribute?	Other Key Attributes that must be unique
Menu = UI Configuration > Menus form	key = Unique Key attribute	Yes	

Menu Items Configuration Unique Identifiers

Attribute Name	Primary Identifier	Editable Attribute?	Other Key Attributes that must be unique
MenuItem = Menu Item form	key = Unique Key	Yes	

MIB Expressions Configuration Unique Identifiers

Attribute Name	Primary Identifier	Editable Attribute?	Other Key Attributes that must be unique
MibExpression = MIB Expression Form	key = Unique Key	Yes	

Monitoring Configuration Unique Identifiers

Attribute Name	Primary Identifier	Editable Attribute?	Other Key Attributes that must be unique
InterfaceSettings = Interface Settings form	uuid	No	ordering = Ordering value
NodeSettings = Node Settings form	uuid	No	ordering = Ordering value

Node Group Configuration Unique Identifiers

Attribute Name	Primary Identifier	Editable Attribute?	Other Key Attributes that must be unique
NodeGroup = Node Group form	uuid	No	name = Name value

Node Group Map Configuration Unique Identifiers

Attribute Name	Primary Identifier	Editable Attribute?	Other Key Attributes that must be unique
NodeGroupMapSettings = Node Group Map Settings Form	uuid	No	

RAMS Server Configuration Unique Identifiers

Attribute Name	Primary Identifier	Editable Attribute?	Other Key Attributes that must be unique
RamsServer = RAMS Server Form	uuid	No	

Security Groups Configuration Unique Identifiers

Attribute Name	Primary Identifier	Editable Attribute?	Other Key Attributes that must be unique
SecurityGroup = Security Group Form	uuid	No	

Security Group Mappings Configuration Unique Identifiers

Attribute Name	Primary Identifier	Editable Attribute?	Other Key Attributes that must be unique
UserToSecurityGroup = Security Group Mappings Form	uuid	No	

NNMi Management Station 6.x/7.x Configuration Unique Identifiers

Attribute Name	Primary Identifier	Editable Attribute?	Other Key Attributes that must be unique
ManagementStation = Management Station Form	uuid	No	

Node Group Status Configuration Unique Identifiers

Attribute Name	Primary Identifier	Editable Attribute?	Other Key Attributes that must be unique
NodeGroupStatusSettings = Node Group Status Settings Form	uuid	No	

User Interface Configuration Unique Identifiers

Attribute Name	Primary Identifier	Editable Attribute?	Other Key Attributes that must be unique
UserInterfaceConfiguration = User Interface Configuration Form	uuid	No	

User Accounts and Roles Configuration Unique Identifiers

Attribute Name	Primary Identifier	Editable Attribute?	Other Key Attributes that must be unique
Account = User Account form	uuid	No	name = Name value
UserGroup = User Group Form	name	yes	
UserGroupMember = User Account Mapping Form	uuid	no	

Back Up and Restore NNMi

As an NNMi administrator, develop a plan for NNMi backups.

For the most complete information, see the "NNMi Backup and Restore Tools" chapter in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals`. See also nnmbackup.ovpl, nnmrestore.ovpl, nnmbackupembddb.ovpl, and nnmrestoreembdb.ovpl (**Help → Documentation Library → Reference Pages**, in the Administrator Commands category).

Use the nnmbackup.ovpl and nnmrestore.ovpl command line tools to do any of the following:

- Back up the NNMi management server and restore data to the same machine.
- Back up the NNMi management server and use the nnmrestore.ovpl command to place the backed up configuration records and database records onto another NNMi management server. For example, moving NNMi to another NNMi management server due to a hardware failure on the original server.

  Note: Both machines must have the same type of operating system and NNMi version and patch level. To move NNMi configuration settings from one computer to another computer that is running a different type of operating system, see "Export and Import Configuration Settings".

  After you restore NNMi on the second NNMi management server, uninstall NNMi from the original NNMi management server. See the *HP Network Node Manager i Software Deployment Reference* for more information.
- Back up the NNMi management server as a safeguard before upgrading the operating system on the server.
- Back up the NNMi management server as a safeguard before updating to a newer version of NNMi.

Note: The back up and restore data might include data from any HP Network Node Manager i Software Smart Plug-ins (iSPIs) installed in your network environment. Check the documentation that came with each NNM iSPI for details.

Use the nnmbackupembddb.ovpl and nnmrestoreembdb.ovpl tools to do the following:

- Back up the NNMi management server embedded database and restore data to the same machine.
- Back up the NNMi embedded database and use the nnmrestoreembdb.ovpl command to place the backed up database records onto another NNMi management server.

The following table summarizes backup and restore tools capabilities:

Command	Backup Embedded DB?	Backup Oracle DB?	Backup other configuration?	Online Backups?	Offline backups?
nnmbackup.ovpl	Yes	No	Yes	Yes	Yes
nnmbackupembddb.ovpl	Yes	No	No	No	Yes

Note the following:

- If you use nnmbackup.ovpl for backup, then use nnmrestore.ovpl to restore the data.
- If you use nnmbackupembddb.ovpl for backup, then use nnmrestoreembdb.ovpl to restore the data.

Before you begin a backup, ensure you have adequate storage space for the backup copy. Verify that you have enough space to store the contents of the directories listed in the following table.

Note: You can compress the files after backup.

See also "About Environment Variables" on page 1578.

NNMi Directories

Operating System	Data	Default Location
Windows Server 2008	Configuration Files	`<`*`drive`*`>:\Program Files (x86)\HP\HP BTO Software` `<`*`drive`*`>` is the drive on which NNMi is installed
	Configuration Data	`<`*`drive`*`>:\ProgramData\HP\HP BTO Software`
	Embedded NNMi Database Storage	`<`*`drive`*`>:\ProgramData\HP\HP BTO Software\shared\nnm\databases\Postgres` If you chose the Oracle database instead of the embedded NNMi database at install time, you must use the Oracle tools for backup in addition to `nnmbackup.ovpl`.
HP-UX, Linux, Solaris	Configuration Files	`/opt/OV`
	Configuration Data	`/var/opt/OV`

NNMi Directories, continued

Operating System	Data	Default Location
	Embedded NNMi Database Storage	`/var/opt/OV/shared/nnm/databases/Postgres` If you chose the Oracle database instead of the embedded NNMi database at install time, you must use the Oracle tools for backup in addition to `nnmbackup.ovpl`.

Related Topics

"Export and Import Configuration Settings" on page 1579

"Archive and Delete Incidents" below

Archive and Delete Incidents

NNMi provides the following options for archiving and deleting incidents:

Auto-trim oldest SNMP trap incident feature

To keep NNMi performing at a high level, NNMi drops incoming SNMP traps (including syslog messages) after storing a specific number of SNMP traps in its database. You can use the auto-trim oldest SNMP trap incidents feature to control the number of SNMP traps (and syslog messages) stored in the NNMi database and to retain important incoming SNMP traps. For more information, see the "Configuring the Auto-Trim Oldest SNMP Trap Incidents Feature" section of the "Maintaining NNMi" chapter in the *HP Network Node Manager i Software Deployment Reference*.

Incident Logging

To ensure that all incidents are archived, use Incident Logging. When using Incident Logging, NNMi logs an incident as soon as it is persisted, even if it is subsequently deleted. If you use the auto-trim oldest SNMP trap incidents feature instead, some incidents might not be archived. For example, if an incident is deleted while Dampened between the specified auto-trim interval, NNMi keeps no record of that incident. For more information about Incident Logging, see "Configure Incident Logging" on page 781.

The `nnmtrimincidents.ovpl` command

NNMi enables you to archive and remove incidents that you no longer want to track. For example, this feature is useful if you want to purge the database of incidents that are older than a specified time period or date. Use the `nnmtrimincidents.ovpl` command to create a comma-separated-values (CSV) file containing the history of incidents, and then trim the volume of incidents to manage the size of your database.

To archive and then delete incidents in NNMi, use the `nnmtrimincidents.ovpl` command. You can choose to only archive or only delete your incidents as described in the arguments table that follows.

> **Note:** By default, NNMi trims incidents without archiving them. To archive incidents before deleting them, use the `-trimAndArchive` option as described in the following

nnmtrimincidents.ovpl Arguments table or use Incident Logging.

Tip: You can also configure NNMi to trim incidents automatically. See the "Reducing the Number of Stored SNMP Trap Incidents" section in the *HP Network Node Manager i Software Deployment Reference*, which is available at: `http://h20230.www2.hp.com/selfsolve/manuals.`

When archiving and deleting incidents, for the best performance results, archive and delete your incidents frequently to keep the size of the NNMi database as small as possible.

SNMP traps are a subset of NNMi incidents (see `-origin` in the arguments table that follows). NNMi monitors the volume of SNMP traps that are stored in the NNMi database. The maximum allowed number of SNMP traps is 100,000. Note the following:

- After 90 percent of the maximum limit for SNMP traps is reached or exceeded, NNMi generates an incident with Severity set to Warning to notify you that NNMi is approaching the maximum limit.
- After 95 percent of the maximum limit for SNMP traps is reached or exceeded, NNMi generates an incident with Severity set to Major to notify you that NNMi is approaching the maximum limit. In addition, NNMi only accepts traps required for Causal Engine analysis until the number of SNMP traps within the database has been reduced using the `nnmtrimincidents.ovpl` command.
- After the maximum SNMP trap limit is reached or exceeded, NNMi generates an incident with Severity set to Critical. NNMi no longer accepts any SNMP traps until the number of SNMP traps within the database has been reduced using the `nnmtrimincidents.ovpl` command.

Use the `nnmtrimincidents.ovpl` command to archive and delete your incidents based on any of the attributes described in the following table. See the nnmtrimincidents.ovpl command for more information, including a complete list of arguments for this command.

Note: The archive's comma-separated-values (CSV) file cannot be used to import the incidents back into NNMi.

nnmtrimincidents.ovpl Arguments

Incident Attribute	Description
-archiveOnly	Specifies that you want to only archive incidents rather than archive and then delete them.
-trimOnly	Specifies that you want to only delete incidents rather than archive and then delete them. **Note:** By default, NNMi trims incidents without archiving them.

nnmtrimincidents.ovpl Arguments, continued

Incident Attribute	Description
-trimAndArchive	Specifies that you want to archive incidents before deleting them.
-date	The date must be entered in the following ISO 8601 format: <*yyyy-mm-dd*>T<*hh*>:<*mm*>:<*ss*>[Z,-<*hh*>:<*mm*>,+<*hh*>:<*mm*>] ISO Date Format: • *yyyy* — Four-digit year • *mm* — Two-digit month • *dd* — Two-digit day • *hh* — Two digits representing the hour (00 through 23) • *mm* — Two digits representing the minutes (00 through 59) • *ss* — Two digits representing the seconds (00 through 59) • +<*hh*>:<*mm*> — Local time zone which is the hours (<*hh*>) and minutes (<*mm*>) ahead of Coordinated Universal Time • -<*hh*>:<*mm*> — Local time zone which is the hours (<*hh*>) and minutes (<*mm*>) behind Coordinated Universal Time For example: `2007-11-05T08:15:30-5:00` corresponds to November 5, 2007, 8:15:30 am, Eastern Standard Time. **Note:** You must specify either a -age or a -date value.
-age	The age of the incident specified in number of days, weeks, or months. **Note:** You must specify either a -age or a -date value.
-family	The incident Family. See Incident Form: General Tab for a list of possible Family values.
-incr	The increment value that helps determine the -age value. Supported increments include **days**, **weeks**, and **months**. The default increment value is **days**.
-path	Specifies the archive file name, including the complete path. The default archive file name is: <*date*> is the date in *yyyy-mm-dd* format <*ms*> is milliseconds **Windows**:

nnmtrimincidents.ovpl Arguments, continued

Incident Attribute	Description

	`%NnmDataDir%\tmp\incidentArchive.<date>.<ms>.txt.gz`
	UNIX:
	`/var/opt/OV/tmp/incidentArchive.<date>.<ms>.txt.gz`
	Note: Each time you generate an archive, NNMi overwrites any existing file with the same name. Therefore, to ensure that all archive files are preserved, provide a unique archive file name each time you want to archive incidents.

-lifecycle	*tional:*Identifies where the incident is in the incident lifecycle. Possible values are **Registered**, **In Progress**, **Completed**, and **Closed**. See About the Incident Lifecycle for more information about **Lifecycle State**.
-name	Identifies the name of the incident configuration.
-nature	*Optional:* Identifies the nature of the incident. Possible values are: **Info, None, Root Cause, Secondary Root Cause, Service Impact,** and **Symptom**. See Using the Incident Form for more information.
-origin	Identifies the Origin of the incident configuration. Possible values are: **Management Software**, **Manually Created**, **Remotely Generated**, and **SNMP Trap**. See Incident Form: General Tab for more information.
-u	The user name required to run this command. This user name must be a valid NNMi user name with a role of either Administrator or System. **Note:** The user name might be a Principal object stored in the NNMi database or might be from Lightweight Directory Access Protocol (LDAP) or X.509 Certificates such as Public Key Infrastructure (PKI) user authentication in your environment. See "Choose a Mode for NNMi Access" on page 504.
-p	The associated password for the user name specified by the -u attribute value. If you do not want to enter an NNMi User Name attribute value and an NNMi Password attribute value at the command line, you can use the nnmsetcmduserpw.ovpl command to specify the valid user name and password (instead of `-u` and `-p`). The credentials set using the `nnmsetcmduserpw.ovpl` command are valid for command execution by the same user. See "Set Up Command Line Access to NNMi" on page 577 for more information.
-quiet	Use this argument when you want to trim incidents without requiring user

nnmtrimincidents.ovpl Arguments, continued

Incident Attribute	Description
	prompts and responses. (Status information appears.)
-sysobjectid	The industry standard SNMP system object ID (RFC 1213, MIB-II `sysObjectID` value that identifies vendor/make/model of a device) assigned to the incident configuration. For SNMP Trap incidents, this value is obtained from the incoming SNMP trap. For Management Event incidents generated by NNMi, the system OID is assigned by NNMi.

For example, delete all incidents with lifecycle equal to Closed and age equal to or greater than 1 month.

```
nnmtrimincidents.ovpl -age 1 -incr months -lifecycle Closed -u
<NNMiadminUsername> -p <NNMiadminPassword>
```

You can also specify a batch size when archiving or deleting incidents. Specify the maximum number of incidents to delete at one time within a single database transaction. This number then determines how often you see a status message that the deletions are complete. Using the default value of 1,000 as an example, NNMi displays a status message after successfully deleting each 1,000 incidents.

> **Note:** The default value of 1,000 was selected to maintain a balance between performance and the frequency of progress messages for the archive and delete operation. This default determines the maximum number of incidents archived and deleted at one time within a single database transaction.

Related Topics

"Back Up and Restore NNMi" on page 1596

Delete Nodes

> **Tip:** To configure NNMi to automatically delete unresponsive nodes, see "Configure Whether to Delete Unresponsive Nodes" on page 212.
>
> To ensure that NNMi never discovers a particular Node in the future, change the Communication Configuration settings, see "Configuring Communication Protocol" on page 119.

Sometimes it is useful to delete Nodes. For example:

- Remove any nodes that are no longer being used in the network.
- Avoid reaching the NNMi license limit for number of managed Nodes by deleting less important Nodes.

- When non-SNMP addresses that had the same DNS hostname are changed to have separate DNS hostnames, NNMi must completely rediscover the non-SNMP nodes to correctly update the database objects (for example, node, interface, address, connection, and incidents).

> **Note:** If you delete a Node with many interfaces and VLANs, you might see an error message indicating that the Node could not be deleted. This means the database was busy with discovery. Try again between discovery cycles.

If a deleted Node is one of your seeds, delete that seed from the Discovery Seeds table as well. See "Delete Discovery Seeds" on page 280.

To understand the results of deleting a Node, click here for more information.

- NNMi cleans up the database by deleting the following objects:
  - Any objects representing a component of the deleted Node (for example, all of that node's interfaces and IP addresses).
  - Any related objects that are empty after deleting the Node (for example, subnets).
  - Any connections with only zero or one end points after deleting the Node.
  - The History of the Node object and all related objects.
- The time required for NNMi to finish deleting depends on the number of objects or related objects being deleted.
- During future discovery cycles, if the deleted Node meets the criteria for an Auto-Discovery Rule and appears in a monitored router's ARP cache, NNMi adds the Node back into the NNMi database during the next discovery cycle. To prevent this, create an Excluded IP Addresses filter for the addresses (see "Configure an Excluded IP Addresses Filter" on page 248).
- During future monitoring cycles, NNMi polls only objects currently in the database.
- Each Incident associated with the deleted Node is modified in the following ways, but not deleted from the NNMi database:
  - The **Status** attribute changes to **Closed**.
  - The **Correlation Notes** indicate the deletion of the associated node, interface, or address.
  - The **RCA State** attribute changes to **FALSE**.

  > **Note:** Incidents generated from SNMP traps or NNM 6.x/7.x Events (received from the deleted Node) appear in the Incident views, but remain unresolved.

- If you are viewing a Node that has recently been deleted by another user, the deleted Node appears as a transparent icon on the map until the map is refreshed using the **Refresh** icon. After **Refresh**, the deleted node is removed from the map. NNMi does not automatically refresh the connectivity or set of nodes in a map view, except on the **Initial Discovery Progress** and **Network Overview** maps.

A subset of NNMi users can delete nodes from a table view, map view, or Node form (depending on the assigned NNMi Role).

> **Note:** By default NNMi Administrators can delete nodes. NNMi Administrators can configure NNMi to permit User Accounts assigned to the NNMi Operator Level 2 User Group to delete nodes. See the *HP Network Node Manager i Software Deployment Reference* for more information (**Help** → **Documentation Library**). Search for "Delete Node".

To delete one or more nodes (maximum 20 at one time):

1. Unmanage the nodes you want to delete.
   a. In a table view, press CTRL-Click and select each row that represents a node you want to unmanage.
   b. Select **Actions** → **Management Mode** → **Unmanage**.

   > **Tip:** You can right-click any object in a table or map view to access the **Actions** menu.

   c. Wait until the Status=*No Status* for each of the following objects:
      - Each Node to be deleted
      - Each Node's Interfaces, IP Addresses, Cards, Ports, and VLAN Ports
2. Do one of the following:
   - *Table views*: Press CTRL-Click and select each row that represents the objects of interest, and click the ✖ Delete icon. Each selected node is deleted from the NNMi database and removed from the current view.
   - *Map views*: click the map symbol representing the node you want to delete, and click **File** → **Delete Node**. The node is deleted from the NNMi database and removed from the current view.
   - *Node form*: select **File** → **Delete Node** and in the confirmation dialog, click **OK**. The form is automatically closed after NNMi deletes the Node.

   > **Note:** If the delete fails, use the nnmnodedelete.ovpl command. Wait for the command to complete.

To delete any number of nodes:

Use the `nnmnodedelete.ovpl` command. See the nnmnodedelete.ovpl Reference Page.

Related Topics

Using Table Views

Using Map Views

Delete One or More Objects

Each row in a table view and each symbol in a map view represents an instance of the object type being displayed. For example, in a node view, each row of the table represents an instance of a node in your network.

Some NNMi users can delete object instances. For example, you might need to delete a node that is no longer being managed. See "Delete Nodes" on page 1602 for more information.

To delete an object instance:

1. Select the object of interest:
   - In a table view, select the row that represents the object.
   - In a map view, click the map symbol.
   - In a form, proceed to step 2.
2. To delete the object, click the ✖ Delete icon.

   The object is deleted from the NNMi database and removed from the current view.

To delete multiple object instances:

1. Select the objects of interest:
   - In a table view, press CTRL-Click and select each row that represents an object you want to delete.
   - In a map view, CTRL-Click each map symbol.
2. To delete the objects, click the ✖ Delete icon.

   > **Note:** For Node objects, you can use this method to delete up to 20 nodes at one time. To delete more than 20 nodes, see the nnmnodedelete.ovpl Reference Page.

   > **Tip:** For all other objects, you can delete any number.

   Each object is deleted from the NNMi database and removed from the current view.

Related Topics

Using Table Views

Using Map Views

"Configure Whether to Delete Unresponsive Nodes" on page 212

Glossary

A

AES

Advanced Encryption Standard

Anycast Rendezvous Point IP Address

Rendezvous Point addresses are loopback addresses used for routers in multi-cast network configurations.

B

BGP

Border Gateway Protocol

C

Causal Engine

The NNMi Causal Engine analyzes the health of your network and provides the ongoing health status reading for each device. The Causal Engine also extensively evaluates problems and determines the root cause for you, whenever possible, sending incidents to notify you of problems. Any incident generated from a Causal Engine management event has an Origin of NNMi in your incident views.

CBC

Cipher Block Chaining

CE

Customer Edge router. The router in your network that sends data to an Internet Service Provider's router (the Provider Edge) on the path to the data's final desination.

Custom Node Collection

A Custom Node Collection identifies a topology node that has at least one associated Custom Poller Policy. Because a topology node can be associated with more than one Policy, the same topology node might appear in multiple Custom Node Collections.

Custom Polled Instance

A Custom Polled Instance represents the results of a MIB expression when it is evaluated against a node. The first time a MIB Expression is validated with discovery information, the results appear in the Monitoring workspace's Custom Polled Instances view. The Custom Polled Instance is updated whenever a change in State occurs and includes the most recent polled value that caused the State to change. These results are then used to determine the Status of the associated Custom Node Collection.

Custom User Groups

Custom User Groups are the User Groups that you create. These User Groups are additional to the NNMi User Groups, which are those User Groups that NNMi provides.

D

DES

Data Encryption Standard

E

EIGRP

Enhanced Interior Gateway Routing Protocol

EVPN

Ethernet Virtual Private Network.

G

global unicast address

(2000:: to 3fff:ffff:ffff:ffff:ffff:ffff:ffff:ffff) A publically routable IPv6 unicast address, used for communication between nodes anywhere on the internet. The first part of the address is a global routing prefix in the 2000::/3 address space for your organization (assigned by the Internet Service Providers). The complete host address can either be manually configured or automatically assigned using IPv6 auto-configuration and neighbor discovery.

H

HMAC

Hash-based Message Authentication Code

hops

A hop is a node representing any network device, such as a workstation, gateway, or switch, which is connected by a link with no intermediate nodes.

HSRP

Hot Standby Router Protocol

I

IPv6 link-local address

A non-routable IPv6 unicast address only used for communication with other nodes on the same link (LAN or VLAN). Link local addresses cannot be used for communication that must be forwarded through a router. IPv6 auto-configuration automatically assigns a unique link local address in the fe80::/10 address space to each IPv6-enabled interface on a system.

ISIS

Intermediate System to Intermediate System Protocol

K

Key Incident

Incidents with both: (1) Severity = other than Normal. (2) Correlation Nature = equal to Root Cause, Service Impact, Stream Correlation, Rate Stream Correlation, Info, or None.

L

Layer 2

Refers to the Data Link layer of the multilayered communication model, Open Systems Interconnection (OSI). The Data Link layer moves data across the physical links in the network. The switches and switch-routers are devices that redirect data messages at the layer 2 level, using the destination Media Access Control (MAC) address to determine where to direct the message.

Layer 3

Refers to the Network layer of the multilayered communication model, Open Systems Interconnection (OSI). The Network layer is concerned with knowing the address of the neighboring nodes in the network, selecting routes and quality of service, and recognizing and forwarding incoming messages to local host domains. The router and switch-router are the devices that redirect data messages at the Layer 3 level. Everything in a subnet is connected at the Layer 3 (IP) level.

Link Aggregation

Protocols used on Switches to configure multiple Interfaces (Aggregation Member Interfaces) to function as if they were one (an Aggregator Interface). When two

Aggregator Interfaces establish a connection, that connection is an Aggregator Layer 2 Connection. The Aggregator Layer 2 Connection appears on Layer 2 Neighbor View maps as a thick line with an Interface icon at each end (representing the Aggregator Interface). When you double-click the thick line, it converts into multiple thin lines representing the participating Aggregation Member Layer 2 Connections with their Aggregation Member Interfaces at each end of the lines.

loopback address

The address associated with the loopback interface. The loopback interface is a virtual interface on a device that provides a route for internal communication. Many vendors provide a specially configured loopback for management purposes. Exact details of how loopbacks are configured varies by vendor and model. See each device's documentation for details. NNMi identifies these loopback addresses by using ifType Number 24, softwareloopback from the IANA ifType-MIB.

M

MAC address

The Media Access Control address (hardware address or physical address) that the factory burns into a network adapter or device with built-in networking capability. A MAC address has six pairs of hexadecimal digits, separated by colons or dashes. For example 02:1F:33:16:BC:55

MAC addresses

The Media Access Control address (hardware address or physical address) that the factory burns into a network adapter or device with built-in networking capability. A MAC address has six pairs of hexadecimal digits, separated by colons or dashes. For example 02:1F:33:16:BC:55

MD5

Message-Digest algorithm 5

MPLS

Multiprotocol Label Switching

multicast address

Used to identify a group of hosts joined into a group. IPv4 multicast addresses are in the range 224.0.0.0 to 239.255.255.255 and IPv6 multicast addresses have the prefix ff00::/8.

multiconnection

A multiconnection is a thick line on a map view between two Node icons, two Node Group icons, or between a Node icon and a Node Group icon (with no Interface icon or IP Address icon at either end of the line). This thick line represents a set of multiple connections that have been combined to preserve space and simplify the map. Your NNMi administrator specifies the number of connections that must exist before NNMi condenses them into a multiconnection line (User Interface Configuration's Multiconnection Threshold attribute). Double-click the thick line to convert it into the original set of connections with Interface icons or IP Address icons at either end of the lines.

N

NAT

Network Address Translation. NNMi supports the following protocols: Static Network Address Translation, Dynamic Network Address Translation, Dynamic Port Address Translation.

NNMi Role

Determined by your membership in one of four special NNMi User Groups (NNMi Administrators, NNMi Level 2 Operators, NNMi Level 1 Operators, or NNMi Guest Users). This membership determines what you can see and do within the NNMi console.

NNMi User Group

NNMi User Groups are those User Groups provided by NNMi. Users cannot access the NNMi console until their User Account is mapped to at least one of the following NNMi User Groups: NNMi Administrators, NNMi Level 2 Operators, NNMi Level 1 Operators (with less access privileges than Level 2 Operators), and NNMi Guest Users

O

OSPF

Open Shortest Path First Protocol

P

PE

Provider Edge router. The Internet Service Provider's router that receives your data on the path to your data's final desination. The Customer Edge (CE) router in your network connects to this PE.

private IP addresses

These are IPv4 addresses that can be reused in home and office local area networks (LANs). Following the standards set by RFC 1918 and RFC 4193 (10.*.*.*, 169.254.*.*, 172.16-31.*.*, and 192.168.*.*)

R

RAMS

HP Router Analytics Management System

routing prefixes

A network protocol technique used to shorten or filter the amount of required routing information in each packet by declaring a prefix for an entire group of packets. This prefix also indicated the number of bits in the address.

S

SHA

Secure Hash Algorithm

U

unique local address

(fd00:: to fdff:ffff:ffff:ffff:ffff:ffff:ffff:ffff) A privately routable IPv6 unicast address used only for communication between nodes within your organization. The unique local addresses cannot be routed to the public internet. The address consists of a routing prefix in the fd00:/8 address spaces, assigned locally by your organization. And the full host address might be manually configured or automatically assigned using IPv6 auto-configuration and neighbor discovery.

Unmanaged

Indicates the Management Mode is "Not Managed" or "Out of Service".

UUID

Universally Unique Object Identifier, which is unique across all databases.

V

VRRP

Virtual Router Redundancy Protocol

Java™
COMPATIBLE