HP Network Node Manager i-series Software

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

Software Version: 8.1x Patch 3

Deployment Guide

Document Release Date: April 2009
Software Release Date: April 2009

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.

Copyright Notices

Trademark Notices

DOM4J® is a registered trademark of MetaStuff, Ltd.

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.

Java™ is a US trademark of Sun Microsystems, Inc.

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

Oracle® is a registered trademark of Oracle Corporation and/or its affiliates.

UNIX® is a registered trademark of The Open Group.

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 USA, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

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

Available Product Documentation

In addition to this guide, the following documentation is available for NNMi:

- *HP Network Node Manager i Software Installation Guide*—Available for each supported operating system on the product media and the NNMi management server.
- *HP Network Node Manager i Software Release notes*—Available on the product media and the NNMi management server.
- *HP Network Node Manager i Software System and Device Support Matrix*—Available on the product media and the NNMi management server.
- *HP Network Node Manager i Software Network Engineering Toolset Guide*—Available on the product media and the NNMi management server.

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

You can visit the HP software support web site at:

www.hp.com/managementsoftware/services

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

HP Software Support Online 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 HP Software 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 find more information about access levels, go to:

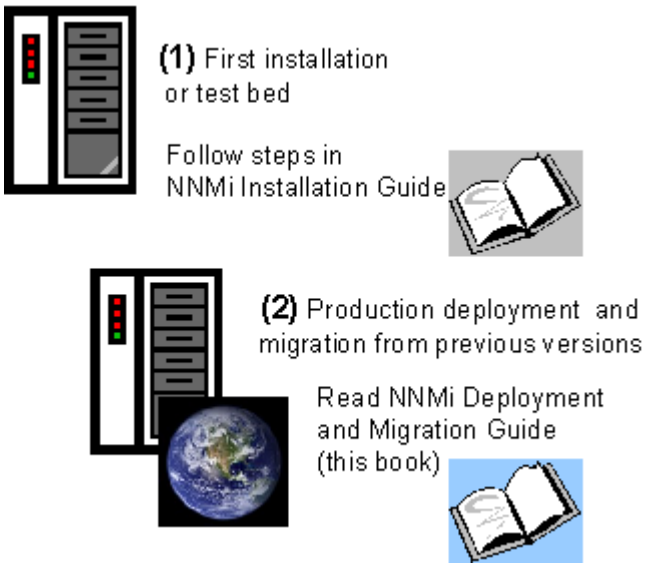
www.hp.com/managementsoftware/access_level

To register for an HP Passport ID, go to:

www.managementsoftware.hp.com/passport-registration.html

Contents

About This Guide



This chapter contains the following topics:

What's in This Guide?

This guide contains a collection of information and best practices for deploying HP Network Node Manager i Software, including NNMi and NNMi Advanced. This guide is written for an expert system administrator, network engineer, or HP support engineer with experience deploying and managing networks in large installations.

This guide assumes that you have already installed NNMi in a limited (test) environment, and that you are familiar with start-up configuration tasks, such as using the Quick Start Configuration wizard to configure community strings, set up discovery for a limited range of network nodes, and create an initial administrator account. To learn more about these tasks, refer to the *NNMi Installation Guide* (see Available Product Documentation on page 3).

HP updates this deployment and migration guide between product releases, as soon as new information becomes available. For information about retrieving an updated version of this document, see Available Product Documentation on page 3.

Environment Variables Used in This Document

This document uses the following two NNMi environment variables to reference file and directory locations. The default values are listed here. Actual values depend upon the selections made during NNMi installation.

- *Windows*:
  - `%NnmInstallDir%`: `<drive>\Program Files\HP\HP BTO Software`
  - `%NnmDataDir%`: `<drive>\Documents and Settings\All Users\Application Data\HP\HP BTO Software`

On Windows systems, the NNMi installation process creates these environment variables so they are always available.

If you first installed NNMi 8.00, your system uses different values for these environment variables, as described in Windows Paths and Environment Variables from NNMi 8.00 on page 336.

- *UNIX®*:
  - `$NnmInstallDir`: `/opt/OV`
  - `$NnmDataDir`: `/var/opt/OV`

On UNIX systems, you must manually create these environment variables if you want to use them.

For information on other NNMi environment variables that you can source, see NNMi Environment Variables on page 333.

Revision History

The following table lists the major changes for each new release of this document.

Document Release Date	Description of Major Changes
NNMi version 8.0x:	
November 2007 (8.00)	First English edition.
February 2008 (patch 1/ 8.01)	• Updated the Windows paths in *NNMi Environment Variables*. • Updated the https information in *Enabling https for NNMi*. • Added to *Discovering Your Network*: — *Best Practice: Save the Existing Configuration* — *Stop Everything and Start Over Again* • Added to *Monitoring Network Nodes*: — *Best Practice: Save the Existing Configuration* — *Monitor Important Interfaces on Unmonitored Nodes* • Added to *Moving NNMi into the Production Environment*: — *Best Practices for Preparing the NNMi Configuration to Be Moved* • Added *Upgrading to NNMi 8.01 from NNMi 8.00*. • Added *NNM iSPI Considerations*.
June 2008 (patch 3/8.02)	• Added to *NNMi Backup and Restore Tools*: — *Limitations of the NNMi Backup and Restore Scripts* • Added *Configuring NNMi in a High Availability Cluster*.
July 2008 (patch 3/8.03)	• Added *Migrating from NNM 6.x/7.x*. • Added *Integration with HP Operations Manager*.
October 2008 (patch 4)	• Updated *Migrating from NNM 6.x/7.x*: — Corrected *Block/Ignore/Disable Traps* • Updated *Integration with HP Operations Manager*.
NNMi version 8.1x:	
November 2008 (8.10)	• Second English edition. Entirely updated. • First Japanese edition.

Document Release Date	Description of Major Changes
December 2008 (patch 1)	All changes are English only. • Updated Enabling https for NNMi for the self-signed certificated provided with NNMi. • Updated Configuring HP NNM i-series Software in a High Availability Cluster for all 8.10 iSPIs. • Updated Upgrading from NNM 6.x/7.x for the migration tools provided in the patch. • Updated Integrating NNM 6.x or NNM 7.x with NNMi 8.11: — Added step 6 in Step 2: Create an NNM 6.x/7.x Management Station Entity in NNMi 8.11. — Significantly revised Generate Test Interface Down and Interface Up Events. • Updated HP Operations Manager: — Added A Normal Situation: Unknown MSI Condition. — Added detail to HPOM Does Not Receive Any Forwarded Incidents.
January 2009 (patch 2/ 8.11)	All changes are English only. • Added Integrating NNMi with a Directory Service through LDAP. • Added Configuring NNMi for Application Failover. • Added Using Single Sign-On with NNMi. • Added HP Business Availability Center. • Added HP Network Automation. • Minor content changes to NNMi Backup and Restore Tools. • Removed *The Causal Engine and NNMi Incidents*. This content is delivered as a separate white paper.
April 2009 (patch 3)	All changes are English only. • Minor content changes to Enabling https for NNMi. • Significantly revised Integrating NNMi with a Directory Service through LDAP. • Updated Configuring NNMi for Application Failover. • Updated Upgrading from NNM 6.x/7.x. • Added Clarus Systems ClarusIPC Plus+. • Updated HP Business Availability Center: • Minor content changes to HP Network Automation. • Minor content changes to HP Operations Manager. • Minor content changes to HP Universal Configuration Management Database.

For More Information about NNMi

To obtain a complete set of information about the NNMi product, use this guide along with other NNMi documentation. The table below shows all NNMi documents to date, including both guides and white papers.

All information below can be downloaded from **http://h20230.www2.hp.com/selfsolve/manuals.** See Available Product Documentation on page 3 for more information.

What do you want to do?	Where to find more information
Install NNMi or NNMi Advanced (first time).	Download the *NNMi Installation Guide*. This guide contains basic steps to install and un-install the product, plus how to do an initial configuration using the NNMi Quick Start Configuration Wizard. • *HP Network Node Manager i-series Software Installation Guide for the Windows Operating System* • *HP Network Node Manager i-series Software Installation Guide for the HP-UX Operating System* • *HP Network Node Manager i-series Software Installation Guide for the Linux Operating System* • *HP Network Node Manager i-series Software Installation Guide for the Solaris Operating System*
Plan for network deployment, including links to system requirements.	See Preparation on page 23 of this guide.
Configure NNMi for a production environment.	See Configuration on page 29 of this guide.
Administer NNMi, including configuration for high availability (HA) or backup.	See Administration on page 75 of this guide.
Migrate to NNMi from previous versions of Network Node Manager.	See Migration from NNM 6.x/7.x on page 191 of this guide.
Learn more about products that integrate with NNMi, including NNMi iSPIs.	See Integrations and Plug-ins on page 255 of this guide.
Install the HP NNMi iSPI Network Engineering Toolset (iNET) and learn about its components.	Download the *HP Network Node Manager i-series Software iSPI Network Engineering Toolset Planning and Installation Guide*.
Obtain documentation for the NNM Developer's Toolkit (SDK).	Download the *NNMi Developer's Toolkit Guide*.

Preparation

This section contains the following chapter:

- Hardware and Software Requirements

Hardware and Software Requirements

This chapter contains the following topics:

Supported Hardware and Software

Before installing NNMi, read the information about NNMi hardware and software requirements that is described in Table 1.

For current versions of all documents listed here, go to:

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

Table 1 Software and hardware pre-installation checklist

Complete (y/n)	Document to Read
	NNMi Installation Guide • **Filename** = `install-guide_en.pdf` • **Windows Media** = DVD main drive (root) • **UNIX Media** = Root directory • **NNMi console** = **Help > NNMi Documentation Library > Installation Guide**
	HP NNMi Software Release Notes • **Filename** = `releasenotes_en.html` • **Windows Media** = DVD main drive (root) • **UNIX Media** = Root directory • **NNMi console** = **Help > NNMi Documentation Library > Release Notes**
	HP NNMi Software System and Device Support Matrix • **Filename** = `supportmatrix_en.html` • **Windows Media** = DVD main drive (root) • **UNIX Media** = Root directory • **NNMi console** = Linked from the release notes

HP updates the *HP NNMi Software System and Device Support Matrix* as new information becomes available. Before you deploy NNMi, check for the most recent NNMi support matrix for your version of the software at:

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

(You must have an HP Passport ID to access this web site.)

If you plan to install Smart Plug-ins (iSPIs), include the system requirements for those products as you plan the NNMi deployment.

System Configuration (UNIX)

If you cannot display NNMi manpages on the NNMi management server, verify that the `MANPATH` variable contains the `/opt/OV/man` location. If it does not, then add the `/opt/OV/man` location to the `MANPATH` variable.

NNMi Coexistence with HP Performance Insight

If you plan to install NNMi on the same server as HP Performance Insight, follow this procedure to avoid problems with the installation sequence and port conflicts:

1 Install HP Performance Insight first.

Do not install NNMi until after you complete step 1 and step 2.

2 Stop all HP Performance Insight processes.

3 Install NNMi. Refer to the *NNMi Installation Guide* for specific instructions.

4 Stop all NNMi processes:

```
ovstop -c
```

5 Modify the `nnm.port.properties` file to resolve any port conflicts. You can find this file in the following directory:

- *Windows*: `%NnmDataDir%\shared\nnm\conf\`
- *UNIX*: `$NnmDataDir/shared/nnm/conf`

6 Start HP Performance Insight processes.

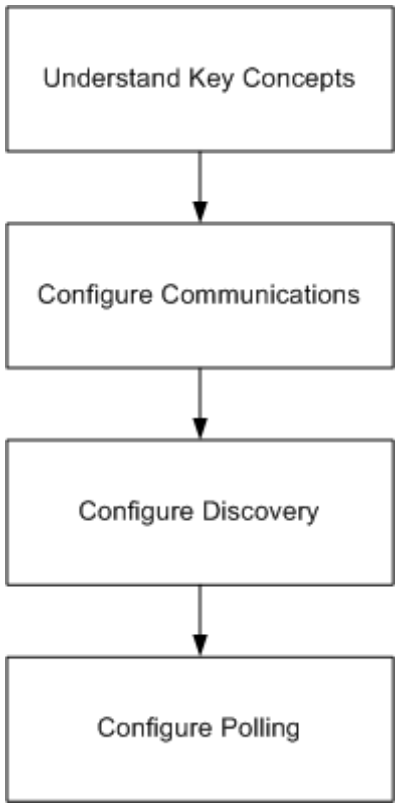
7 Start all NNMi processes:

```
ovstart -c
```

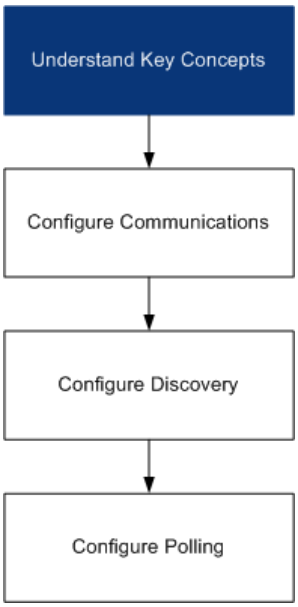
Configuration

This section contains the following chapters:

- General Concepts for Configuration
- NNMi Communications
- NNMi Discovery
- NNMi State Polling



General Concepts for Configuration



Read this chapter for an introduction to concepts that are explained in more detail later in this guide. This chapter also contains some best practices that apply to all HP Network Node Manager i Software configuration areas.

This chapter contains the following topics:

Task Flow Model

The chapters in the configuration section of this guide support the following task flow:

1. **Concepts**—Gain a general understanding of the configuration area. The information in this guide supplements the information in the NNMi help.
2. **Plan**—Decide how you want to approach the configuration. This is a good time to begin or update your company's network management documentation.
3. **Configure**—Use a combination of the NNMi console, configuration files, and command line interface to enter the configuration into NNMi. Refer to the NNMi help for specific procedures.
4. **Evaluate**—In the NNMi console, examine the results of your configuration. Adjust the configuration as necessary to achieve the desired results.
5. **Tune**—Optional. Adjust the configuration to improve NNMi performance.

Best Practice: Save the Existing Configuration

It is a good idea to save a copy of the existing configuration before you make any major configuration changes. If you do not like the results of your configuration changes, it is easy to revert to your saved configuration.

Use the `nnmconfigexport.ovpl` command to save the current configuration. To recover a saved configuration, use the `nnmconfigimport.ovpl` command.

For information on how to use these commands, refer to the appropriate reference pages, or the UNIX manpages.

Best Practice: Use the Author Attribute

The following configuration forms include the **Author** attribute:

- Device Profile
- Incident Configuration
- URL Action

As you create or modify the configurations on these forms, set the **Author** attribute to a value that identifies your organization. When you export the NNMi configuration, you can specify an author value to pull only those items that your organization has customized.

User Interface Model

The NNMi console uses a transaction approach to updating the database. The changes that you make in the NNMi console forms do not take effect until you save and close the forms all of the way back to the NNMi console.

The **Discovery Seed** form is an exception. This form is provided on the **Discovery Configuration** form as a convenience, but it is disconnected from the rest of discovery configuration. For this reason, you must save and close the **Discovery Configuration** form to implement your auto-discovery *rules* before you configure any discovery seeds for those rules.

Ordering

Some NNMi console configuration forms include the **Ordering** attribute, which sets the priority for applying the configurations. For one configuration area, NNMi evaluates each item against the configurations from the smallest (lowest) ordering number to the next lowest ordering number, and so on, until NNMi finds a match. At that point, NNMi uses the information from the matching configuration and ceases to look for any more matches. (The communication configuration is an exception. NNMi continues to search for information to complete the communication settings.)

The **Ordering** attribute plays an important role in NNMi configuration. If you see unexpected discovery or status results, check the ordering of the configurations for that area.

The following configurations use the **Ordering** attribute:

- **Regions** on the **Communication Configuration** form
- **Auto-Discovery Rule** on the **Discovery Configuration** form
- **Interface Settings** on the **Monitoring Configuration** form
- **Node Settings** on the **Monitoring Configuration** form
- (NNMi Advanced) **Configuration Per Node Group**, which is part of trap or event configuration on the **Incident Configuration** form

Ordering numbers are also used in the following places, but with different meanings:

- Ordering on the **URL Action** form sets the order of items on the **Actions** menu.
- Ordering on the **Node Group Map Settings** form sets the order of items in the **Topology Maps** workspace.

For specific information about how the **Ordering** attribute affects a given configuration area, refer to the NNMi help for that area.

Best practice

For each configuration area, apply low ordering numbers to the most restrictive configurations, and apply high ordering numbers to the least restrictive configurations.

Best practice

For each configuration area, all ordering numbers must be unique. During initial configuration use ordering numbers with a standard interval to provide flexibility for future modifications to the configuration. For example, give the first three configurations the ordering numbers 100, 200, and 300.

Node Groups

NNMi's primary filtering technique is by grouping nodes or interfaces, and then applying settings to a group or filtering visualizations by group. Node groups can be used for any or all of the following purposes:

- Monitoring settings
- Table filtering
- Customizing map views

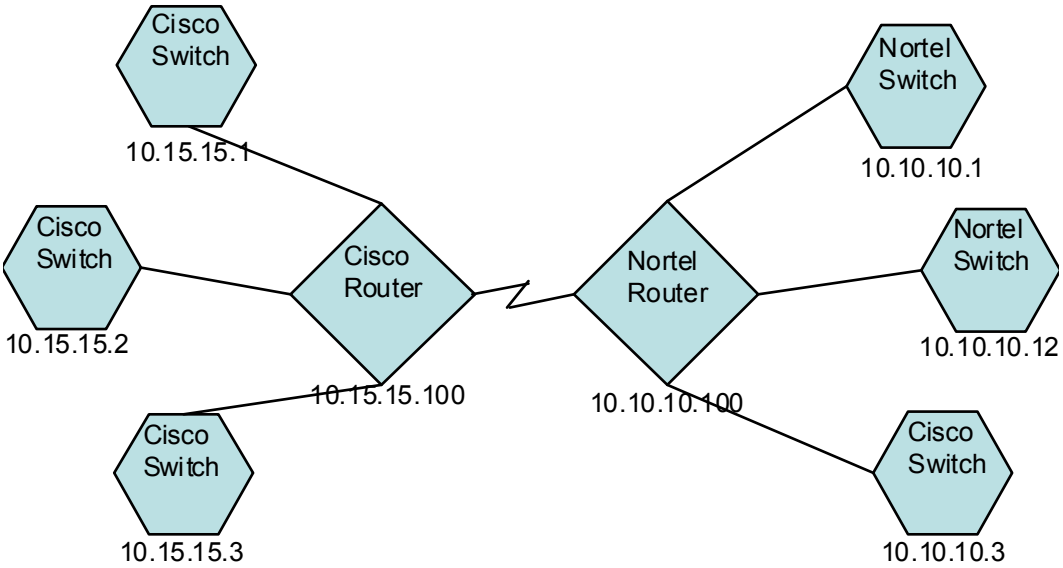
Interface groups can be used for either or both of the following purposes:

- Monitoring settings
- Table filtering

You can create a hierarchy of node groups based on any filterable attribute(s) to control map view drill-down and/or monitoring setting inheritance.

Group Overlap

Regardless of the way(s) that you intend to use group definitions, the first step is to define which nodes or interfaces are members of a group. Because you can create groups for different purposes, each object can be included in multiple groups. Consider the following example:



- For monitoring purposes, you might want to set a polling interval of 3 minutes for all switches, regardless of vendor or location. You can do this with a device category filter.
- For maintenance purposes, you might want to group all Cisco switches so that you can place them OUT OF SERVICE together for IOS upgrades. You can do this with a vendor filter.
- For visualization, you might want to group all devices on the 10.10.*.* site into a container with propagated status. You can do this with an IP address filter.

The Cisco switch with IP address 10.10.10.3 would qualify for all three groups.

You want to find the balance between having a usably rich set of groups available for configuration and viewing, and overloading the list with superfluous entries that will never be used.

Node Groups

Node group membership is evaluated by comparing each discovered node to each of the groups configured. You can use four types of information to filter which nodes qualify:

- Additional nodes that you specify explicitly

OR'd with

- Inclusion of child groups

OR'd with

  - Device filters based on capabilities derived from the system object ID

  AND with

  - Additional filters based on other attributes

Any specific node that you add is a member of the group. A member of a child group qualifies as a member of this group. For other nodes, they must match criteria on the **Device Filters** tab AND'ed with the **Additional Filters** tab. Within the **Additional Filters** tab you can create complex logic of AND, OR, and parenthetical evaluation.

Hierarchies/Containment

You can create simple, reusable, atomic groups and combine them hierarchically for monitoring or visualization. Using hierarchical containers for nodes greatly enhances map views by providing cues about the location or type of object at fault. NNMi gives you complete control of the definition of the groups and their drill-down order.

You can create simple, reusable atomic groups first, and then specify them as child groups as you build up. Alternatively, you can specify your largest parent group first and create child groups as you go.

For example, a network might contain Cisco switches, Cisco routers, Nortel switches, and Nortel routers. You can create parent groups for Cisco devices and for all switches. Because the hierarchy is specified when you create the parent and designate its children, each child group, such as Cisco switches, can have multiple parents.

Hierarchies work well for the following situations:

- Types of nodes with similar monitoring needs
- Geographical locations of nodes
- Types of nodes to be taken OUT OF SERVICE together
- Groups of nodes by operator job responsibility

When you use groups in map views and table views, you see a (configurable) propagated status for the group.

Keep in mind that as you use group definitions to specify monitoring configuration, hierarchy does *not* imply ordering for settings. The settings with the lowest ordering number apply to a node. By carefully incrementing ordering numbers, you can emulate inheritance concepts for settings.

The configuration interface automatically prevents circular hierarchy definitions.

Device Filters

During discovery, NNMi collects direct information through SNMP queries and derives other information from that through device profiles. (For more information, see NNMi Derives Attributes through Device Profiles on page 50.) By gathering the system object ID, NNMi can index through the correct device profile to derive the following information:

- Vendor
- Device category
- Device family within the category

These derived values, in addition to the device profile itself, are available for use as filters.

For example, you can group all objects from a specific vendor, regardless of device type and family. Or you can group all devices of a type such as router, across vendors.

Additional Filters

With the additional filters editor, you can create custom logic to match fields including:

- hostname (Hostname)
- mgmtIPAddress (Management Address)
- hostedIPAddress (Address)
- sysName (System Name)
- sysLocation (System Location)
- sysContact (System Contact)
- capability (Unique Key of the Capability)
- customAttrName (Custom Attribute Name)
- customAttrValue (Custom Attribute Value)

Filters can include the `AND`, `OR`, and grouping (parentheses) operations. For more information, refer to *Specify Node Group Additional Filters* in the NNMi help.

Capabilities are primarily intended for other programs that integrate with NNMi. For example, router redundancy and component health add capabilities (fields) to the NNMi database. You can view these capabilities by examining the node details from a device that has already been discovered.

Custom attributes can be added by iSPIs, or you can create your own custom attributes. If you have not purchased the Web Services SDK, you must place values in the field for each node manually. For example, an asset number or serial number might be an attribute that is not a capability.

Additional Nodes

If the network contains critical devices that are too difficult to qualify using filters, you can add them to a group by individual hostname.

Interface Groups

Interface groups filter interfaces within nodes by IFType or by other attributes, such as ifAlias, ifDescr, ifName, ifIndex, IP address, and so forth. Interface groups carry no hierarchy or containment, although you can further qualify membership based on the node group for the node hosting the interface.

Interface groups can be filtered on custom capabilities and attributes similarly to node groups.

Qualifications for interface groups are AND'd together within and across tabs.

Node/Interface/Address Hierarchy

NNMi assigns monitoring settings in the following manner:

1. **Interface Settings**—NNMi monitors each of the node's interfaces and IP addresses based on the first matching **Interface Settings** definition. The first match is the **Interface Settings** definition with the lowest ordering number.
2. **Node Settings**—NNMi monitors each node and each previously unmatched interface or IP address based on the first matching **Node Settings** definition. The first match is the **Node Settings** definition with the lowest ordering number.

   Child node groups are included in the ordering hierarchy. 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 step 1 or step 2, NNMi applies the default monitoring configuration settings.

Stop Everything and Start Over Again

If you want to completely restart discovery and redo all of the NNMi configuration, or if the NNMi database has become corrupted, you can reset the NNMi configuration and database. This process deletes *all* of the NNMi configuration, topology, and incidents.

For information about the commands identified in this procedure, see the appropriate reference pages, or the UNIX manpages.

Follow these steps:

1. Stop the NNMi services:

   ```
   ovstop -c
   ```
2. Optional. Because this procedure deletes the database, you might want to back up the existing database before proceeding:

   ```
   nnmbackup.ovpl -type offline -target <backup_directory>
   ```

3. Optional. If you want to keep any of the current NNMi configuration, use the `nnmconfigexport.ovpl` command to output the NNMi configuration to an XML file.
4. Optional. Use tho `nnmtrimincidents.ovpl` command to archive the NNMi incidents.
5. Drop and recreate the NNMi database.
   - For the embedded database, run the following command:

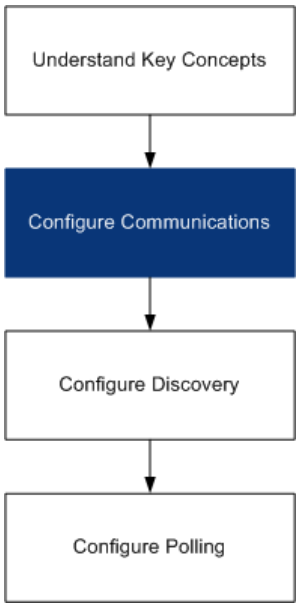
     ```
     nnmresetembdb.ovpl -nostart
     ```

   - For an Oracle database, ask the Oracle database administrator to drop and recreate the NNMi database. Maintain the database instance name.
6. If you have installed iSPIs or stand-alone products that integrate with NNMi, reset those products to remove the old topology identifiers. For specific procedures, see the product documentation.
7. Start the NNMi services:

   ```
   ovstart -c
   ```

   NNMi now has only the default configurations as if you had just installed the product on a new system.
8. Start configuring NNMi. Do one of the following:
   - Use the Quick Start Configuration Wizard.
   - Enter information into the **Configuration** workspace in the NNMi console.
   - Use the `nnmconfigimport.ovpl` command to import some or all of the NNMi configuration that you saved in step 3.

NNMi Communications



HP Network Node Manager i Software uses both SNMP and ICMP (ping) protocols to discover devices and to monitor device status and health. To establish viable communication in your environment, you configure NNMi with the access credentials and appropriate timeout and retry values for different devices and areas of your network. You can disable a protocol in some areas of your network to reduce traffic or to respect firewalls.

The communication values that you configure form the foundation of NNMi's discovery and state polling. NNMi applies the configured values to all device queries for discovery or polling data. Thus, if you disable SNMP communication to some region of your network, neither discovery nor polling will be able to send SNMP traffic to that region or receive data back.

This chapter contains the following topics:

Concepts for Communications

NNMi uses SNMP and ICMP in a primarily request-response manner. The management station initiates the communication looking for simple responsiveness (ICMP ping) or requesting specific data values as measured by the device (SNMP MIB objects).

The following concepts apply to communications configuration:

- Preferred Management Address
- Polling Protocols

Network Latency and Timeouts

For either protocol, normal network latency influences the turnaround time for the management station to get its answer. Different areas of your network customarily have different turnaround times for the answer to come back. The local network where the management station resides will have almost instantaneous response. A response from a device in a remote geographical region accessed through a dial-up wide area link would typically take much longer to receive.

In addition, devices which are heavily loaded may be too busy to respond to queries right away. This latency must be added to the network turnaround itself.

You configure how long NNMi should wait for an answer and how many times it should try, based on what is normal for each area of your environment and perhaps each device or type of device.

For each retry, NNMi adds the configured timeout value to the previous timeout value.

Access Credentials and Community Strings

SNMP communication requires a form of access control called a community string (or user in SNMPv3), which acts much like a password. Each request from the management station contains the community string and the managed device responds only if the community string is correct.

In SNMPv1 and SNMPv2, community strings pass through the network in clear text and are not really very secure.

With SNMPv3, security becomes a major objective. Each user configured carries a profile to allow authentication (proving that the packet really originated from the stated user through MD5 or SHA) and optionally privacy (encryption through DES). The communication from the manager must not only contain the correct user, but also follow the configured authentication and privacy controls for that user.

NNMi allows you to configure multiple community strings or users that may be in use for a group of nodes. It attempts communication with a device in the group by trying all configured values (at a given security level, see below) in parallel. The first value to return a response from the device becomes the credential in use for discovery and monitoring. If a credential stops working (you reconfigured the device), NNMi retries the configured values to find the one that works now.

By default, if there is no SNMP response from the device for any configured credentials, the device is not discovered and added to the database. However, you can override this default, in which case the device might be discovered with the `No SNMP` device profile.

SNMP Version Preferences

The SNMP protocol itself has evolved over the years from version 1 to version 2(c) and now version 3, with increasing security capabilities (among others). NNMi can handle any or a mix of all versions in your environment.

As part of your configuration, you inform NNMi of the minimum level of security acceptable in each area of your network. If you set security constraints to the minimum level, NNMi selects the preferred version of the protocol by doing the following. The first response at any point sets the communication credentials and version to be used and evaluation stops.

1. If there are one or more community strings configured, attempt to communicate using SNMPv2 with the configured values for timeouts and retries.
2. If there is no response to any community string using SNMPv2, attempt to communicate using SNMPv1 with the configured values for timeouts and retries.
3. If there are one or more SNMPv3 users configured:
   a. If there are users with No Authentication, No Privacy, attempt to communicate using SNMPv3 with the configured values for timeouts and retries.
   b. If there are users with Authentication, No Privacy, attempt to communicate using SNMPv3 with the configured values for timeouts and retries.
   c. If there are users with Authentication, Privacy, attempt to communicate using SNMPv3 with the configured values for timeouts and retries.
4. By default, if there is no SNMP response from the device for any configured credentials, the device is not discovered and added to the database. However, you may override this and the device may be discovered with the `No SNMP` device profile.

Configuration Levels

NNMi allows you to configure access credentials, timeouts and retries, and enablement of protocols at the following levels:

- Specific node
- Regions
- Global Defaults

At each level you may configure access credentials, timeout and retry values, and protocol enablement. You may also leave settings blank at one level and have the next level of defaults applied.

When NNMi looks to communicate with a given IP address, NNMi applies the configuration settings as follows:

1. If the address matches a specific node configuration, fill in all available values. Leave blanks where the settings have blanks.
2. If there are still blanks, evaluate the address to see if it matches a region definition. Because regions may overlap, use the match with the lowest ordering number. Fill in any blanks which have values available at the regional level.
3. If there are still blanks, fill them from the global default settings. (Additional region matches are not used.)

So the values to apply are built up cumulatively by continuing to examine higher level defaults until all blanks are filled.

Preferred Management Address

Devices such as routers have many IP addresses assigned to them. If you are using auto-discovery rules, NNMi discovers the router by the presence of any one of its IP addresses in some other device's ARP cache. NNMi then discovers all of the router's interfaces and selects one address to use as the **management IP address** for all further communication with the device. For information about how NNMi determines the management IP address, refer to *Node Form* in the NNMi help.

If NNMi's selected management address does not align with your preferences, you can override it, forcing NNMi to use the address that you configure.

Best practice

One of NNMi's rules specifies using the first discovered IP address as the management address. You can influence NNMi's choosing by configuring your preferred IP as the seed address. For suggestions regarding seed selection, see Discovery Seeds on page 55.

Polling Protocols

You can disable access to a portion of your network for SNMP or ICMP if a firewall will block access.

Disabling ICMP traffic to an area of the network has the following results:

- The auto-discovery rule ping sweep feature cannot locate additional nodes. All nodes must either be seeded or available through ARP cache or discovery protocol (for example, CDP or EDP) MIB entries. Wide area networking devices might be missed unless you seed every one of them.
- State polling cannot monitor devices for which there is no SNMP access. By default, this is the only time the State Poller uses ICMP, although you can configure ICMP fault polling groups.
- Operators cannot use **Actions > Ping** to check device reachability during troubleshooting.

Disabling SNMP traffic to an area of the network has the following results:

- Discovery cannot gather any information about the devices except that they exist. All devices will have the `No SNMP` device profile.
- Discovery cannot learn of additional devices through queries. All devices must be directly seeded.
- Discovery cannot gather connectivity information from your devices, so they will appear unconnected.
- State Poller uses the configuration for `No SNMP` devices and defaults to using ICMP (ping) to monitor the availability of the devices.
- State Poller cannot gather component health or performance data from devices.
- Causal Engine cannot contact devices to perform neighbor analysis and locate the root cause of incidents.

Plan Communications

Make decisions in the following areas:

- Meaningful Defaults
- Communication Configuration Regions
- Specific Node Configurations
- Retry and Timeout Values
- Active Protocols
- Multiple Community Strings or Access Credentials

Meaningful Defaults

Because the default values will be used to fill any blanks at the end of evaluation, set them to be reasonable for the majority of your network.

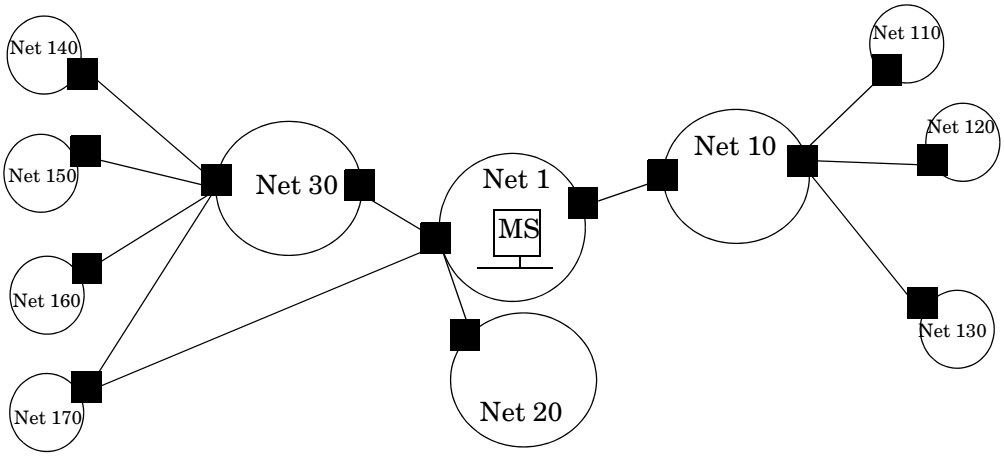
- Do you have commonly-used community strings that NNMi should try?
- If no other settings apply, what timeout and retry values are reasonable in your network?

Other values are configured at the regional or specific node level.

Communication Configuration Regions

Regions represent areas of your network where similar communication settings make sense. For example the local network around the management stations returns responses very quickly. Areas of your network that are multiple hops away typically take longer to respond.

You do not need to configure each subnet or area of your network; you can combine them based on similar lag times. Consider the following network map:



For timeout/retry purposes, you may want to configure a region for Net 1; a region to include Net 10, Net 20, and Net 30; and a region for the more distant outlying networks. You would decide how best to group Net 170, depending on whether traffic management configuration is set to prefer the one-hop or two-hop path from the management server.

Regions are also used to group devices which use similar access credentials. If all routers in your network use the same community string (or small set of possible strings) and you can identify them with a naming convention such as `rtr`*`nnn`*`.yourdomain.com`, you can configure a region so that they are handled similarly. If you cannot use a wildcard to group the devices, you can configure them as specific nodes.

Now you can plan your region configuration so that you will be able to apply the same timeout/retry value and access credential configurations to all nodes in a region.

Region definitions may overlap and a device may qualify for multiple regions. NNMi applies the settings from the lowest order number region.

Specific Node Configurations

For any devices which do not match regional definitions due to special parameters or lack of wildcard ability, document that you will need a specific node configuration and the values to be used for the node.

Retry and Timeout Values

Configuring longer timeouts and more retries can garner more responses from devices that are busy or distant. This eliminates false down messages. However, it also lengthens the time to determine that actual down devices require attention. Finding the balance for each area of your network is important and may require a period of testing and adjusting values in your environment.

To get an idea of current lag time for each hop, run a `traceroute` to a device in each network area.

Active Protocols

You have two opportunities to control the type of traffic going to a device: at the communication layer and in the monitoring settings. Use the communication layer when firewalls in your infrastructure prohibit types of traffic. Use monitoring settings when you do not need that type of data for a set of devices. If either layer disables a protocol, that traffic will not be sent to the device.

Note whether each region or specific device should receive ICMP traffic.

Disabling SNMP communication significantly compromises NNMi's ability to monitor the status and health of your network.

You do not need to explicitly disable SNMP communication with devices for which you do not supply access credentials. By default, NNMi assigns those devices to the `No SNMP` device profile and monitors them using ICMP only.

Multiple Community Strings or Access Credentials

For the global defaults and for each region, you may configure multiple community strings to be tried in parallel.

While trying probable community strings, NNMi may cause devices to generate authentication failures. Inform your operations department that authentication failures may safely be ignored while NNMi completes its initial discovery.

Alternatively, you can minimize the number of authentication failures by configuring your regions and the community strings to try as tightly as possible.

If your environment uses both SNMPv1/2 and SNMPv3, determine the minimum acceptable security level for each region as well as the credentials to try at each level.

Plan the access credentials to be tried for each area of your network.

Configure Communications

After reading the information in this section, refer to *Configuring Communication Protocol* in the NNMi help for specific procedures.

It is a good idea to save a copy of the existing configuration before you make any major configuration changes. For more information, see Best Practice: Save the Existing Configuration on page 32.

Configure the following areas of communication:

- Default settings
- Region definitions and their settings
- Specific node settings

  For specific nodes, you can enter node settings through the NNMi console or through a configuration file.

Save and Close all **Communication Configuration** forms all of the way back to the NNMi console to implement your changes.

Best practice

Double-check your order numbers, keeping in mind that an object which qualifies for multiple groups has settings applied from the group with the lowest order number.

Evaluate Communications

This section lists ways to evaluate the progress and success of the communications settings. Most of these tasks can be completed only after discovery has completed.

Consider the following:

- Are All Nodes Configured for SNMP?
- Is SNMP Access Currently Available for a Device?
- Is NNMi Using the Correct Communications Settings?

Are All Nodes Configured for SNMP?

1. Open the **Nodes** inventory view.
2. Filter the **Device Profile** column to contain the string `No SNMP`.
   - For each of the devices that you want to manage, configure communication settings for the specific node. Alternatively, you can expand a region to include the node and update the access credentials.
   - If the communication settings are correct, verify that the SNMP agent on the device is running and properly configured (including ACLs).

Is SNMP Access Currently Available for a Device?

1. Select the node in an inventory view.
2. Select **Actions > Status Poll** or **Actions > Configuration Poll**.

   If the results show any SNMP values, communication is operational.

You can also test communication from the command line with the `nnmsnmpwalk` command. For more information, refer to the *nnmsnmpwalk* reference page, or the UNIX manpage.

Is NNMi Using the Correct Communications Settings?

Missing or incorrect SNMP community strings can result in incomplete discovery or can negatively impact the discovery performance.

To verify the settings configured for a device, use the `nnmcommconf.ovpl` command or follow these steps:

1. Select the node in an inventory view.
2. Select **Actions > Communication Settings**.

   NNMi evaluates the specific node matching, regional settings by order number, and default settings to arrive at the displayed values.

Example output of the `nnmcommconf.ovpl` command follows:

```
SNMP Configuration for node1.domain.com
name = node1.domain.com
address = 10.2.100.6
addressForced = false
getCommunity = [public]
timeout = 5000
retries = 1
port = 161
enabled = true
region name = Core Network
```

This output shows the current timeout and retry values, the IP address for communication, and, if you have administrator access, the configured access credentials.

Verify that NNMi's values for `getCommunity` and `region name` are correct. The get community string should match the read-only community string that the SNMP agent uses. The region name should match the name of the auto-discovery rule that you expect this node to match.

To determine the values actually in use from the most recent discovery of the node:

1. Open a **Node** form.
2. At the **SNMP Agent** field, open the **SNMP Agent** form.

   The form shows the protocol version and other SNMP settings.

Do the State Poller Settings Agree with the Communication Settings?

Even if the communication settings allow protocol traffic to an area of your network, that type of traffic might be disabled in the monitoring settings. To determine whether the settings are being overridden:

1. Select the node in an inventory view.
2. Select **Actions > Monitoring Settings**.

If either the Monitoring Settings or the Communication Settings disable a type of traffic to the device, that traffic will not be sent from NNMi.

Is the Management IP Address Correct?

1. Select the node in an inventory view.
2. Select **Actions > Communication Settings**.

Tune Communications

Reduce authentication failures

If NNMi is generating too many authentication traps during discovery, configure smaller regions or specific nodes with smaller groups of access credentials for NNMi to try.

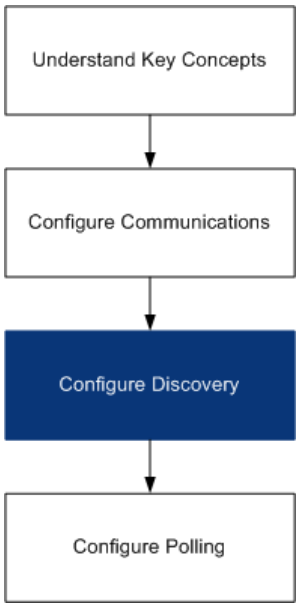
Tune timeouts and retries

When NNMi attempts to contact a device using SNMP during discovery, the communication configuration determines NNMi's ability to gather the necessary device information. When the communication configuration does not include the correct SNMP community strings, or if NNMi is discovering non-SNMP devices, NNMi uses the configured settings for SNMP timeouts and retries. In this case, large timeout values or a high number of retries can negatively impact the overall performance of discovery. If your network contains devices that you know respond slowly to SNMP/ICMP requests, consider using the **Regions** or **Specific Node Settings** tabs on the **Communication Configuration** form to fine tune the timeout and retry values for just these devices.

Reduce default community strings

Having a large number of default community strings can negatively impact discovery performance. Instead of entering many default community strings, fine tune the community string configuration for particular areas of your network by using the **Regions** or **Specific Node Settings** tabs on the **Communication Configuration** form.

NNMi Discovery



One of the most important network management tasks is keeping your view of the network topology current. HP Network Node Manager i Software discovery populates the topology inventory with information about the nodes in your network. NNMi maintains this topology information through ongoing Spiral Discovery, which ensures that root cause analysis and the troubleshooting tools provide accurate information regarding incidents.

This chapter provides information to help you configure NNMi discovery. For an introduction to how discovery works and for detailed information about how to configure discovery, see *Discovering Your Network* in the NNMi help.

If you have experience working with NNM 6.x/7.x and you want to understand how discovery has changed in NNMi 8.11, see Network Discovery on page 193 for a high-level overview of the differences.

This chapter contains the following topics:

Concepts for Discovery

The NNMi default behavior of discovering only routers and switches allows you to focus your network management on the critical or most important devices. In other words, target the backbone of the network first. Generally, you should avoid managing end nodes (for example, personal computers or printers) unless the end node is identified as a critical resource. For example, database and application servers might be considered critical resources.

NNMi provides several ways to control what devices to discover and include in the NNMi topology. Your discovery configuration can be very simple, quite complex, or anywhere in between, depending on how your network is organized and what you want to manage with NNMi.

NNMi does not perform any out-of-the-box discovery. You must configure discovery before any devices appear in the NNMi topology.

Each discovered node counts toward the license limit, regardless of whether NNMi is actively managing that node. The capacity of your NNMi license might influence your approach to discovery.

Status monitoring considerations might also influence your choices. By default, the State Poller only monitors interfaces connected to devices NNMi has discovered. You can override this default for some areas of your network, and you can discover the devices beyond the edge of your responsibility. (For information on the State Poller, see NNMi State Polling on page 61.)

NNMi provides two primary discovery configuration models:

- **List-based discovery**—Explicitly tell NNMi exactly which devices should be added to the database and monitored through a list of seeds.
- **Rule-based discovery**—Tell NNMi which areas of your network and device types should be added to the database, give NNMi a starting address in each area, and then let NNMi discover the defined devices.

You can use any combination of list-based and rule-based discovery to configure what NNMi should discover. Initial discovery adds these devices to the NNMi topology, and then spiral discovery routinely rediscovers the network to ensure that the topology remains current.

NNMi Derives Attributes through Device Profiles

As NNMi discovers devices, it uses SNMP to gather some attributes directly. One of the key attributes is the MIB II system object ID (`sysObjectID`). From the system object ID, NNMi derives additional attributes, such as vendor, device category, and device family.

During discovery, NNMi collects the MIB II system capabilities and stores them in the topology portion of the database. System capabilities are visible on the **Node Details** form. However, these capabilities are not used by any other portion of NNMi (specifically, monitoring configuration). NNMi uses the device category (from the device profile for the system object ID) to match devices into node groups. In node view tables, the **Device Category** column identifies the device category for each node.

NNMi ships with over 3600 device profiles for system object IDs that were available at the time of release. You can configure custom device profiles for the unique devices in your environment to map these devices to category, vendor, and so forth.

Plan Discovery

Make decisions in the following areas:

- Select Your Primary Discovery Approach
- Auto-Discovery Rules
- Node Name Resolution
- Subnet Connection Rules
- Discovery Seeds
- Rediscovery Interval
- Do Not Discover Objects

Select Your Primary Discovery Approach

Decide whether to do entirely list-based discovery, entirely rule-based discovery, or a combination of both approaches.

List-Based Discovery

With list-based discovery, you explicitly specify (as a discovery seed) each node that NNMi should discover.

Benefits of using only list-based discovery include:

- Provides very tight control over what NNMi manages.
- Simplest configuration.
- Good for fairly static networks.
- A good way to start using NNMi. You can add auto-discovery rules over time.

Disadvantages of using only list-based discovery include:

- NNMi does not discover new nodes as they are added to the network.
- You must provide the complete list of nodes to be discovered.

Rule-Based Discovery

With rule-based discovery, you create one or more auto-discovery rules to define the areas of the network that NNMi should discover and include in the NNMi topology. For each rule, you must provide one or more discovery seeds (by explicitly naming seeds or by enabling ping sweep), and then NNMi discovers the network automatically.

Benefits of using rule-based discovery include:

- Good for large networks. NNMi can discover a large number of devices based on minimal configuration input.
- Good for networks that change frequently. New devices that are added to the network are discovered without administrator intervention (assuming that each device is covered by an auto-discovery rule).

- Ensures that any new device added to your network is discovered to comply with service level agreements for managing new devices in a timely manner or security guidelines to flag unauthorized new devices.

Disadvantages of using rule-based discovery include:

- It is easier to run into license limitations.
- Depending on the structure of your network, tuning auto-discovery rules can be complex.
- If auto-discovery rules are very broad and NNMi discovers many more devices the than you want to manage, you might want to delete the unneeded devices from NNMi topology. NNMi does not provide a tool for bulk deletion, so it is very difficult to clean up from a very broad discovery.

Rule-based discovery only

Auto-Discovery Rules

Auto-Discovery Rule Ordering

The value of an auto-discovery rule's **Ordering** attribute affects discovery ranges in the following ways:

- IP address ranges

  If a device falls within two auto-discovery rules, the settings in the auto-discovery rule with the lowest ordering number applies. For example, if an auto-discovery rule excludes a set of IP addresses, then no other auto-discovery rules with higher ordering numbers process those nodes and the nodes within that range of addresses are not discovered unless they are listed as discovery seeds.

- System object ID ranges

  — If no IP address range is included in an auto-discovery rule, then the system object ID settings apply to all auto-discovery rules with higher ordering numbers.

  — If an IP address range is included in an auto-discovery rule, the system object ID range applies only within the auto-discovery rule.

Exclude Devices from Discovery

- To prevent discovery of certain object types, create an auto-discovery rule with a low ordering number that ignores the system object IDs that you do not want discovered. Do not include an IP address range in this rule. By giving this auto-discovery rule a low ordering number, the discovery process quickly passes by the objects that match this rule.
- The **Ignored by rule** setting for an IP address range or a system object ID range affects that auto-discovery rule only. The devices included in an ignored range are available to be included in another auto-discovery rule.
- New in NNMi 8.10. The addresses listed on the **Excluded IP Addresses** tab of the **Discovery Configuration** form apply to all auto-discovery rules. Unless they are configured as discovery seeds, these addresses are never added to the NNMi topology. (Discovery seeds are always discovered.)

Ping Sweep

As of NNMi 8.10, you can use ping sweep to locate devices within the IP address ranges of the configured auto-discovery rules. For initial discovery, you might want to enable ping sweep for all rules. Doing so provides enough information to NNMi discovery that you do not need to configure discovery seeds.

➤ Ping sweep works for subnets of 16 bits or smaller, for example, `10.10.*.*`.

Ping sweeps are especially useful for discovering devices across a WAN that you do not control, such as an ISP network.

➤ Firewalls often view ping sweeps as attacks on the network, in which case, a firewall might block all traffic from a device that emits ping sweeps.

Best practice Enable ping sweep for small discovery ranges only.

Discovery Seeds for Auto-Discovery Rules

Provide at least one discovery seed per auto-discovery rule. The options for providing the seeds are as follows:

- Enter seeds on the **Discovery Seeds** tab of the **Discovery Configuration** form.
- Use the `nnmloadseeds.ovpl` command to load information from a seed file.
- Enable ping sweep for the rule, at least for initial discovery.

Best Practices for Auto-Discovery Rules

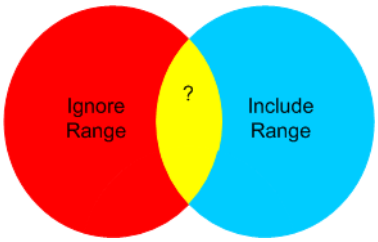
- Because NNMi automatically manages all discovered devices, use IP address ranges that closely match the areas of the network that you want to manage.
  - — You can use multiple IP address ranges within an auto-discovery rule to restrict discovery.
  - — You can add a large IP address range to an auto-discovery rule and then exclude some IP addresses from discovery within that rule.
- The system object ID range specification is a prefix, not an absolute value. For example, the range 1.3.6.1.4.1.11 is the same as 1.3.6.1.4.1.11.*.

Examples

Discovery Rule Overlap

Figure 1 shows two discovery ranges that overlap. The circle on the left represents an IP address range or a system object ID range to be ignored by NNMi discovery. The circle on the right represents an IP address range or a system object ID range to be discovered and included in the NNMi topology. The overlapping region might be included or ignored by discovery, depending on the ordering of these auto-discovery rules.

Figure 1 Overlapping Discovery Ranges



Limit Device Type Discovery

To discover all HP devices in your network that are not printers, create one auto-discovery rule with a range to include the HP enterprise system object ID (1.3.6.1.4.1.11). In this auto-discovery rule, create a second range to ignore the system object IDs of HP printers (1.3.6.1.4.1.11.2.3 9). Leave the IP address range unset.

Node Name Resolution

By default, NNMi attempts to identify a node in the following order:

1. Short DNS name
2. Short sysName
3. IP Address

The following scenarios describe situations in which you might want to change the default order for node name resolution:

- If your organization is dependent on others to update the DNS configuration, you might set a policy of defining the sysName for each new device as it is added to the network. In this case, set select sysName as the first choice for node name resolution so that NNMi can discover the new device as soon as it is deployed in the network. (Maintain the sysName over the life of the device.)
- If your organization does not set or maintain the sysName for managed devices, select sysName as the third option for node name resolution.

Best practice

If you use the full or short DNS name as the primary naming convention, confirm that you have forward and reverse DNS resolution from the NNMi management server to all managed devices.

When the full DNS name is the naming convention, labels on the topology maps can be long.

Best practice

As of NNMi 8.10, NNMi selects the lowest loopback address as the management address for Cisco devices, so put DNS resolution on the lowest loopback address for each Cisco device.By contrast, NNMi 8.0x selects the highest loopback address as the management address.

Subnet Connection Rules

List-based discovery only

For list-based discovery, NNMi uses the subnet connection rules to detect connections that span a WAN. NNMi evaluates the subnet membership of the device it has discovered on each end of a probable connection (by examining their IP addresses and subnet prefixes) and looks at subnet connection rules for a match.

Rule-based discovery only

When auto-discovery rules are enabled and NNMi finds a device configured with a subnet prefix between /28 and /31:

1. NNMi checks for an applicable subnet connection rule.
2. If a match is found, NNMi uses each valid address in the subnet as a hint and attempts a discovery on that address.

Best practice

Use the default connection rules. Only modify them if you have a problem.

Discovery Seeds

List the devices to use as discovery seeds.

Best practice

One of NNMi's rules for selecting the preferred management IP address specifies using the first discovered IP address as the management address. You can influence NNMi by configuring the preferred IP address as the seed address.

Best practice

For Cisco devices, use a loopback address as the discovery seed because loopback addresses are more reliably reachable than other addresses on a device. Ensure that DNS is correctly configured to resolve the device hostname to the loopback address.

List-based discovery only

For list-based discovery, list all devices that you want NNMi to manage. You might be able to export this list from asset management software or from some other tool.

Because NNMi does not automatically add any devices to this list, ensure that the list includes every device for which you have responsibility or which influences your monitoring and status calculations.

Rule-based discovery only

As of NNMi 8.10, discovery seeds are optional for rule-based discovery:

- If ping sweep is enabled for an auto-discovery rule, you do not need to specify a seed for that rule.
- For each auto-discovery rule with ping sweep disabled, identify at least one seed per rule. If a rule includes multiple IP address regions, you might need a seed in each routable region because routers do not keep ARP entries across WAN links.

Best practice

For the most complete rule-based discovery, use routers, not switches, as discovery seeds because routers generally have much larger ARP caches than do switches. A core router connected to a network that you want to discover is an excellent choice for a discovery seed.

Rediscovery Interval

NNMi rechecks the configuration information from each device in the database according to the configured rediscovery interval. In addition, NNMi collects the ARP cache from each router covered by an auto-discovery rule and looks for new nodes on the network.

Any change in the communication-related configuration of a device, such as interface renumbering, automatically triggers NNMi to update its data for that device and its neighbors.

The following changes do not trigger an automatic rediscovery; devices are updated only at the configured rediscovery interval:

- Changes within a node (for example, firmware upgrade or system contact).
- New nodes added to the network.

Select the rediscovery interval to match the level of change in the network. For a highly-dynamic network, you might want to use the minimum interval of 24 hours. For more stable networks, you can safely extend that period.

Do Not Discover Objects

In NNMi, there are three ways that you can configure NNMi to disregard certain objects:

- On the **Communication Configuration** form, you can turn off ICMP and/or SNMP communication at various levels: globally, for communication regions, or for specific hostnames or IP addresses. For information on the impacts of disabling one or both of these protocols, see Polling Protocols on page 42.
- On the **Discovery Configuration** form, you can set up an auto-discovery rule that instructs NNMi to never gather hints from certain IP addresses or SNMP system object IDs. Nodes matching the criteria still appear on the map and in the database, but spiral discovery does not extend to the neighboring devices beyond those IP addresses or object types.
- On the **Discovery Configuration** form, you can set up an auto-discovery rule that instructs NNMi to exclude certain IP address ranges and/or specific IP addresses from the database. Spiral discovery does not display those addresses on any node's list of addresses or use those addresses when establishing connections between devices, so NNMi never monitors the health of those addresses.

Configure Discovery

This section lists configuration tips and provides some configuration examples. After reading the information in this section, refer to *Configure Discovery* in the NNMi help for specific procedures.

Because NNMi launches discovery from seeds as soon as you **Save and Close** the **Discovery Seed** form, ensure that you do the following before you configure seeds:

- Complete all communication configuration.
- Complete all auto-discovery rules (if any).
- Configure subnet connection rules.
- Configure name resolution preferences.
- **Save and Close** all the way back to the console.

It is a good idea to save a copy of the existing configuration before you make any major configuration changes. For more information, see Best Practice: Save the Existing Configuration on page 32.

Tips for Configuring Auto-Discovery Rules

- As you define a new auto-discovery rule, check each setting carefully. For a new rule, auto-discovery is enabled by default, IP address ranges are included by default, and system object ID ranges are *ignored* by default.

Tips for Configuring Seeds

- If you already have a file that lists the nodes to be discovered, format this information as a seed file and use the `nnmloadseeds.ovpl` command to import the node list into NNMi.
- In the seed file, specify IP addresses as a way of influencing the IP address that NNMi chooses as the management address. (If you use hostnames, DNS provides the IP address for each node.)
- A good format for each entry in the seed file is shown here:

  ```
  IP_address # node name
  ```

  This format is easy for both NNMi and human readers.
- For maintenance purposes, it is better to use only one seed file. Add nodes as needed and then rerun the `nnmloadseeds.ovpl` command. NNMi discovers the new nodes but does not re-evaluate the existing nodes.
- Removing a node from the seed file does not remove it from the NNMi topology. Delete the node directly in the NNMi console.
- Deleting a node from a map or inventory view does not delete the seed.
- If you want NNMi to rediscover a node, delete that node from a map or inventory view *and* from the **Discovery Seeds** tab on the **Discovery Configuration** form in the NNMi console, and the re-enter the node in the NNMi console, or run the `nnmloadseeds.ovpl` command.

Rule-based discovery only

- Completely configure a discovery rule *before* you specify a seed for that rule. That is, click **Save and Close** on the **Discovery Configuration** form. (Although it appears to be connected, the **Discovery Seeds** tab is a separate form that is not part of the **Discovery Configuration** form in the database model. As a result, when you save the information on the **Discovery Seeds** tab, NNMi updates the seed configuration immediately.)

Evaluate Discovery

This section lists ways to evaluate the progress and success of discovery.

Follow the Progress of Initial Discovery

NNMi discovery is dynamic and ongoing; it is never complete, so you will never see a "discovery completed" message. The process of initial discovery and connection takes some time. The following items suggest ways to gauge the progress of initial discovery:

- In the **About Network Node Manager i-series** window, watch for the node count to reach the expected level and stabilize. This window does not refresh automatically. During initial discovery, open the **About Network Node Manager i-series** window several times.
- On the **Discovery Configuration** form, look at the **Discovery Seeds** tab. Refresh this tab until all seeds show the `Node created` results, which indicates that the device has been added to the topology database. This result does *not* indicate that NNMi has gathered all information from the device and processed its connectivity.
- Open the **Node** form for representative nodes. When the **Discovery State** field transitions to `Discovery Completed`, NNMi has gathered the node's basic characteristics as well as the node's ARP cache and discovery protocol neighbors, if applicable. This state does *not* indicate that NNMi has completed connectivity analysis for the device.
- In the **Nodes** inventory view, scan to see that key devices are present from different areas of your network.
- Open the **Layer 2 Neighbors** view for representative nodes to determine whether connectivity analysis has completed for that area.
- Review the **Layer 2 Connections** and **VLANs** inventory views to gauge the progress of layer 2 processing.

Were All Seeds Discovered?

1. Open the **Discovery Configuration** form.
2. On the **Discovery Seeds** tab, sort the list of nodes by the **Discovery Seed Results** column. For any node in an error state, consider the following:
   - Failed discovery due to an unreachable node or unresolved DNS name—For these types of failures, verify network connectivity to the node and check for accurate DNS name resolution. To work around DNS issues, use the IP address to seed the node or include the hostname in a `hostNoLookup.conf` file.
   - License node count exceeded—This scenario occurs when the number of devices already discovered reached your license limit. You can either delete some discovered nodes or purchase additional node pack licenses.
   - Node discovered but no SNMP response—SNMP communication problems can occur for seeded devices as well as devices that are discovered through auto-discovery. For more information, see Evaluate Communications on page 45.

Do All Nodes Have a Valid Device Profile?

1. Open the **Nodes** inventory view.
2. Filter the **Device Profile** column to contain the string `No Device Profile`.
3. If a node is discovered but has no device profile, add a new device profile in the **Configuration > Device Profiles** view, and then perform a configuration poll on the node to update its data.

Were All Nodes Discovered Properly?

Examine the data in the **Nodes** inventory view. If any nodes do not have a management address, check the communication settings for those nodes as described in Are All Nodes Configured for SNMP? on page 46.

If any expected nodes are missing from the **Nodes** inventory view, check the following:

- On each missing node, verify that the discovery protocol (for example, CDP) is correctly configured.
- If a missing node is on a WAN, enable ping sweep for the auto-discovery rule that includes that node.

List-based discovery only

Auto-Discovery Rules

If you see unexpected discovery results, re-evaluate the auto-discovery rules.

When NNMi discovery finds an address hint, it uses the first matching rule to determine if a node should be created. If no rules are matched, NNMi discovery discards the hint. The ordering number for auto-discovery rules determines the order in which the auto-discovery rule configuration settings are applied.

For each auto-discovery rule, check the following settings:

- **Enable Auto-Discovery** must be enabled for auto-discovery to occur for the rule.
- Verify that the following settings are correct for the type of nodes you want discovered for the rule:
  - **Discover Any SNMP Device**
  - **Discover Non-SNMP Devices**

  Remember that only routers and switches are discovered by default and non-SNMP nodes are *not* discovered. Enabling these settings without considering your environment can result in NNMi discovering more nodes than intended.

IP Address Ranges

The IP address of a discovery hint must match an **Include in Rule** entry in the IP address range list. If there are no included IP address ranges in an auto-discovery rule, then all address hints are considered a match. (For this case, see Tips for Configuring Auto-Discovery Rules on page 57.) Additionally, the hint must *not* match any entry marked **Ignored by Rule**. If all checks successfully match, this rule's configuration is used for handling the hint.

- If you are not discovering some expected devices, check your configured IP ranges to ensure that the IP addresses for those devices are included in a range and not ignored by a rule with a lower ordering number.

- If you are discovering more devices that you want, modify the include ranges or add ignored ranges for the IP addresses of the devices that you do not want discovered. Also, determine if **Discover Any SNMP Devices** is enabled.

System Object ID Ranges

The system object ID (OID) from a discovery hint must match an **Include in Rule** entry in the system object ID ranges list. If there are no included system object ID ranges in an auto-discovery rule, then all object IDs are considered a match. Additionally, the OID must not match any entry marked **Ignored by Rule**. If all checks successfully match, this rule's configuration is used for handling the hint.

- Use the system object ID ranges to either expand auto-discovery to include more than the default routers and switches, or to exclude specific routers and switches.
- Each node must match both the IP address range and the system object ID range specified before it is discovered and added to the topology database.

Are All Connections and VLANs Correct?

NNMi creates Layer 2 connections and VLANs as a separate step after devices are added to the topology. Give NNMi plenty of time for initial discovery before evaluating connections and VLANs.

To evaluate Layer 2 connectivity, create a node group for each network area of interest, and then display a topology map for that node group. (In the **Node Groups** inventory, select a node group, and then click **Actions > Node Group Map**.) Look for any nodes that are not connected to the other nodes in this map.

To evaluate VLANs, from the **VLANs** inventory view, open each **VLAN** form, and then examine the list of ports for that VLAN.

Rediscover a Device

1. Perform a configuration poll of the device.
2. Delete the device.

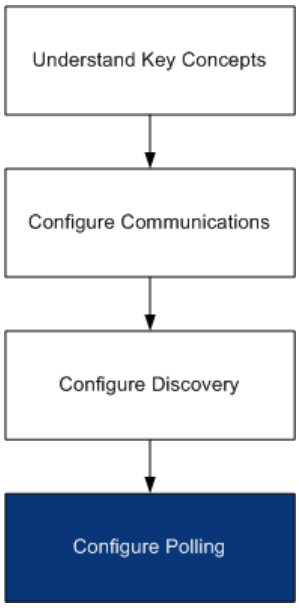
If the device is a seed, delete the seed, and then re-add the seed.

Tune Discovery

For general discovery performance, fine tune the discovery configuration to discover only critical and important devices.

- Filter by IP address range and/or system object ID.
- Limit discovery of non-SNMP devices and any SNMP devices (devices that are not switches or routers).

NNMi State Polling



This chapter provides information to help you expand and fine tune network monitoring by configuring the HP Network Node Manager i Software State Poller service. This chapter supplements the information in the NNMi help. For an introduction to how monitoring works and for detailed information about how to configure monitoring, refer to *Monitoring Network Health* in the NNMi help.

If you have experience working with NNM 6.x/7.x and you want to understand how monitoring has changed in NNMi 8.11, see Status Monitoring on page 196 for a high-level overview of the differences.

This chapter contains the following topics:

Concepts for State Polling

This section provides a brief overview of network monitoring, including the order that the State Poller uses to evaluate polling groups. After reading the information in this section, continue to Plan State Polling on page 62 for more specific information.

As with network discovery, you should focus network monitoring on the critical or most important devices in the network. NNMi can only poll devices in the topology database. You control which network devices NNMi monitors, the type of polling to use, and the interval at which to poll.

You can use the interface and node settings on the **Monitoring Configuration** form to refine the status polling of devices, and to set different polling types and intervals for different classes, types of interfaces, and types of nodes.

You can configure State Poller data collection to be based on an ICMP (ping) response, or to be based on SNMP data. NNMi automatically handles the mapping from the type of data collection you enable to the actual MIB objects internally, significantly simplifying configuration.

As you plan polling configuration, you should carefully consider how to set up interface groups and node groups for the State Poller service. If you are new to the concept of *groups*, see Node Groups on page 33, and Node/Interface/Address Hierarchy on page 37 for overview information.

Order of evaluation

Because an interface or node may qualify for multiple groups, the State Poller applies the configured polling interval and polling type in a well-defined order of evaluation. For each object in the discovered topology:

1. If the object is an interface, State Poller looks for a qualifying interface group. Groups are evaluated from the lowest Order Number to the highest. The first matching group is used and evaluation stops.
2. If no interface group has captured the object, node groups are evaluated from lowest Order Number to highest. The first matching group is used and evaluation stops. Any contained interface which has not qualified for an interface group on its own characteristics inherits the polling settings from its hosting node.
3. For devices that are discovered but not included in any node or interface settings definitions, the global monitoring settings (on the **Default Settings** tab of the **Monitoring Configuration** form) establish the monitoring behavior.

Plan State Polling

This section provides information to plan for State Poller configuration, including a polling configuration checklist; and more detailed information to help you plan for monitoring, decide how to create polling groups, and determine what types of data should be captured during the polling process.

Polling Checklist

You can use the checklist below to plan for State Poller configuration.

- ☐ What can be monitored?
- ☐ What are the logical groups for monitored items, based on object type, location, relative importance, or other criteria?
- ☐ How often should each grouping be monitored?
- ☐ What data should be collected to capture information about the monitored item? This may include:
  - — ICMP (ping) response
  - — SNMP fault data
  - — SNMP performance data if you have the NNM iSPI for Performance license
  - — Additional SNMP Component Health data

Example polling configuration

To help you understand the polling configuration process, consider this example. Suppose that your network contains the latest proxy servers from ProximiT. You need to ensure that these devices can be reached, but you do not require SNMP monitoring of the proxy servers.

1 What can be monitored?

Because you can only monitor what has been discovered, you configure auto-discovery rules to ensure that NNMi's database contains your ProximiT proxy servers. For more information on configuring discovery, see NNMi Discovery on page 49.

2 What are the logical groups for monitored items?

Obviously you'd like to group the ProximiT proxy servers together and apply the same monitoring settings to all of them. Because you are not doing interface (SNMP) monitoring for the devices, you do not need any interface groups.

You can also use this node group to filter views, to check the status of the proxy servers as a group, and to put the group OUT OF SERVICE to update firmware.

3 How often should each group be monitored?

For your service level agreements, a five minute polling interval for the proxy servers is sufficient.

4 What data should be collected?

Here's where the monitoring configuration differs from other groups. For our ProximiT proxy server example, you enable ICMP fault monitoring and disable SNMP fault and polling monitoring. Without SNMP fault monitoring for the group, Component Health monitoring will not apply.

For more detailed planning information concerning these configuration choices, see the following topics:

- What Can Be Monitored? on page 63
- Planning Groups on page 65
- Planning Polling Intervals on page 67
- Deciding What Data to Collect on page 68

What Can Be Monitored?

By default, the NNMi State Poller uses SNMP polls to monitor the following:

- Interfaces that are connected to another known interface on an NNMi-discovered device.
- Router interfaces which host IP addresses.

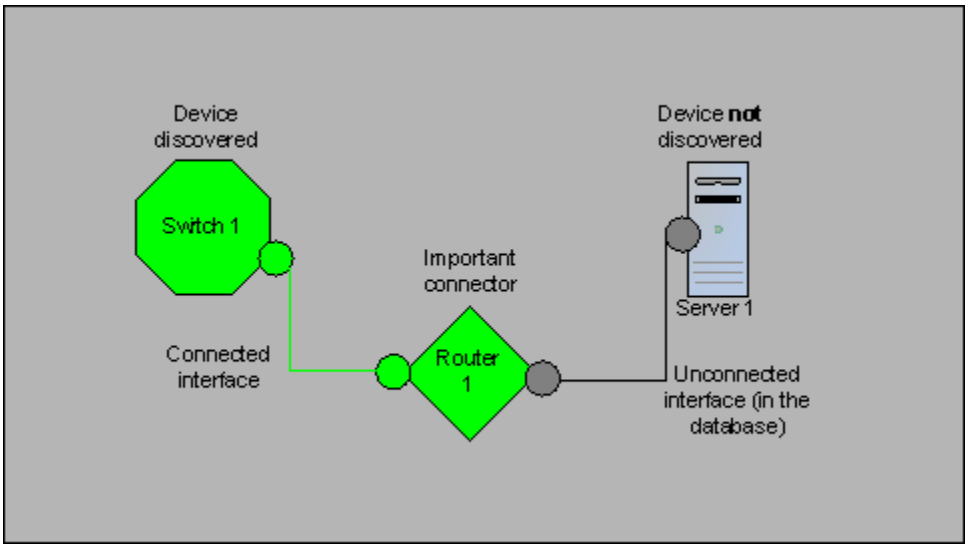
In most cases, polling only connected interfaces provides sufficiently accurate root-cause analysis. Extending the set of monitored interfaces can impact polling performance.

Extend monitoring

You can extend the monitoring to include the following:

- Unconnected interfaces. By default, the only unconnected interfaces that NNMi monitors are those that have IP addresses *and* are included in the **Routers** node group.

  NNMi defines an unconnected interface as an interface that is not connected to another device discovered by NNMi, as shown below.



- Interfaces, such as router interfaces, that have an IP address.
- ICMP polling for devices that do not support SNMP. By default, ICMP polling is enabled for the **Non-SNMP Devices** node group.

Interfaces to Unmonitored Nodes

Sometimes, you need to know the status of an interface that connects to a device you do not manage directly. For example, you want to know whether the connection to an application or Internet server is up, but you might not be responsible for maintaining that server. If you do not include the server in the discovery rules, NNMi sees the interface that faces the server as unconnected.

There are two ways to monitor the status of an important interface that connects to an unmonitored node.

- Discover the unmonitored node

  When you add an unmonitored node to the NNMi topology, NNMi sees the interfaces connecting the node to the rest of the topology as CONNECTED. Then NNMi can poll these interfaces according to the monitoring configuration. NNMi discovers the node as MANAGED. Unmanage nodes that you do not want NNMi to monitor.

  Each discovered node counts toward the license limit, regardless of whether NNMi is actively managing that node.

- Poll the unconnected interface

  You can create a node group containing the network devices that provide connectivity for undiscovered nodes. Then enable polling of unconnected interfaces for the node group.

  NNMi polls *all* interfaces on the devices in the node group, which can add a lot of traffic for a device with many interfaces.

Stop Monitoring

The NNMi management modes are used to set devices or interfaces to UNMANAGED or OUT OF SERVICE. UNMANAGED is considered to be a permanent situation; you will never care to know the status of the object. OUT OF SERVICE is for temporary situations where the object(s) will be offline and down incidents would be superfluous.

Consider the management mode as an overlay across all group settings. Regardless of its group, polling interval, or type, the State Poller does not communicate with an object when its status is set to UNMANAGED or OUT OF SERVICE.

Best practice

Some of the devices and/or interfaces you choose to discover and place in the database do not need to be polled. Note those objects which you will permanently set to UNMANAGED. You may want to create one or more node groups to allow you to set management modes more easily.

Planning Groups

You must set up node and interface groups before configuring monitoring settings. Therefore, you must consider polling requirements while configuring node and interface groups. Ideally, node and interface groups are configured so that you can monitor important devices frequently, and you can check on non-critical devices less frequently (if at all).

These groups are defined through the **Configuration > Node Groups** or **Configuration > Interface Groups** work spaces and are, by default, the same groups that are used to filter incident, node, interface, and address views. To create a separate set of node or interface filters for configuring monitoring settings, open a node or interface group and select the **Add to View Filter List** check box on the **Node Group** or **Interface Group** form. Click **Save and Close**.

You can set polling types and polling intervals at a node group or interface group level on the **Node Settings** and **Interface Settings** tabs of the **Monitoring Configuration** form.

Determine the criteria by which you want to group interfaces and/or devices by similar polling needs. Here are some factors to consider in your planning:

- Which area of your network contains these devices? Are there timing constraints?
- Do you want to differentiate polling intervals or data gathered by device type? By interface type?
- Does NNMi provide pre-configured groups you can use?

Best practice

You can create group definitions for objects that are likely to go OUT OF SERVICE at the same time, whether by location or some other criteria. For example, you could put all your Cisco routers into OUT OF SERVICE mode while you apply an IOS upgrade.

Interface Groups

Based on your criteria, determine which Interface groups to create. Remember that interface groups are evaluated first (see Concepts for State Polling on page 61). Interface groups can reference node group membership, so you may end up configuring node groups before interface groups to implement your plan.

Preconfigured interface groups

NNMi has several useful interface groups already configured for you to use. These include:

- All interfaces with an IFType related to ISDN connections
- Interfaces for voice connections
- Interfaces for point-to-point communication
- Software loopback interfaces
- VLAN interfaces
- Interfaces participating in link aggregation protocols

Over time HP may add more default groups to simplify your configuration tasks. You can use existing groups, modify them, or create your own.

Interface groups have two types of qualifiers: node group membership for the hosting node and IFType or other attribute for the interface. You can choose to combine these as follows:

- All interfaces on nodes in a node group are grouped regardless of IFType; do not select any IFTypes or attributes (name, alias, description, speed, index, address, etc.).
- All interfaces of certain IFTypes or set of attributes are grouped, regardless of the node on which they reside.
- Only interfaces of a certain IFType or attributes that reside on a particular group of nodes are grouped.

Node Groups

After planning interface groups, plan node groups. Not all node groups created for monitoring make sense for filtering views, so you can configure them independently.

Preconfigured node groups

HP provides a default collection of node groups to simplify your configuration tasks. These are based on device categories derived from the system object ID during the Discovery process. The node groups provided by default include:

- Routers
- Networking Infrastructure Devices (switches, routers, etc.)
- Microsoft Windows Systems
- Devices for which you do not have the SNMP community string
- Important Nodes. This is used internally by the Causal Engine to provide special handling for devices in the "shadow" of a connector failure. For more information, refer to *Node Groups As Predefined View Filters* in the NNMi help.

Over time HP may add more default groups to simplify your configuration tasks. You can use existing groups, modify them, or create your own.

You can qualify the definition of related nodes using the following node attributes:

- IP address(es) on the node
- Hostname wildcard convention
- Device Profile derivatives such as category, vendor, and family
- MIB II sysName, sysContact, sysLocation

Best practice

You can create simple, reusable, atomic groups and combine them into hierarchical clusters for monitoring or visualization. Group definitions can overlap, such as "All Routers" and "All systems with IP address ending in .100." Nodes will probably qualify for multiple groups as well.

Find a balance by creating a rich set of groups for configuration and viewing without overloading the list with superfluous entries that will never be used.

Interaction with Device Profiles

When each device is discovered, NNMi uses its system object ID to index into the list of available Device Profiles. The Device Profile is used to derive additional attributes of the device, such as vendor, product family, and device category.

As you configure node groups, you can use these derived attributes to categorize devices to apply monitoring settings. For example, you may want to poll all switches regardless of vendor throughout your network on a certain polling interval. You can use the derived device category, Switch, as the defining characteristic of your node group. All discovered devices whose system object ID maps to the category, Switches, will receive the configured settings for the node group.

Planning Polling Intervals

For each object group, you select a polling interval that NNMi uses to collect data. The interval can be as short as one minute, or as long as days in order to best match your Service Level Agreements.

Best practice

Shorter intervals help you become aware of network problems as soon as possible; however, polling too many objects in too short an interval can cause a backlog in the State Poller. Find the best balance between resource utilization and intervals for your environment.

Deciding What Data to Collect

The State Poller service uses polls to gather state information about the monitored devices in your network. Polling can be done using ICMP and/or SNMP.

ICMP (ping) ICMP address monitoring uses ping requests to verify the availability of each managed IP address.

SNMP SNMP monitoring verifies that each monitored SNMP agent is responding to SNMP queries.

- The State Poller is highly optimized to collect configured SNMP information from each monitored object with one query at each interval. When you save configuration changes, the State Poller recalculates the group membership of each object and reapplies the configured interval and set of data to collect.
- SNMP monitoring issues SNMP queries for all monitored interfaces and components, requesting the current values from the MIB II interface table, the HostResources MIB, and vendor-specific MIBs. Some values are used for fault monitoring. If you have the NNM iSPI for Performance installed, some are used for performance measurement.

SNMP Component Health data You may enable or disable Component Health monitoring at the global level. Component Health monitoring for faults follows the fault polling interval settings for the device.

Gathering additional data at each poll does not affect the time to execute the poll. However, additional data stored for each object can increase the memory requirements for State Poller.

Performance monitoring settings are only used with the NNM iSPI for Performance. Component Health monitoring for performance follows the performance polling interval settings for the device.

Best practice Batching your monitoring configuration changes is less disruptive to State Poller ongoing operation.

Configure State Polling

This section provides configuration tips and provides some configuration examples. After reading the information in this section, refer to *Configure Monitoring Behavior* in the NNMi help for specific procedures.

It is a good idea to save a copy of the existing configuration before you make any major configuration changes. For more information, see Best Practice: Save the Existing Configuration on page 32.

Configure Interface Groups and Node Groups

You create interface groups and node groups in the **Configuration** workspace. For more information, refer to *Creating Groups of Nodes or Interfaces* in the NNMi help.

Examples

For example, to configure a node group for ProximiT proxy servers:

1. Open **Configuration > Node Groups** and click **New**.
2. Name the group **`Proxy Servers`** and check **Add to View Filter List**.
3. Select the **Hostname Wildcards** tab and click **New**.
4. Enter the wildcard as **`prox*.yourdomain.com`** and click **Save and Close**.

   If you had configured a Device Profile and Category for the ProximiT devices, you could use the **Device Filters** tab to access the **Device Category** selector and base the group on the Proxy Server category you created.
5. Click **Save and Close** on the group definition.

You must configure node groups before you can reference them in your interface group configuration.

Configure Interface Monitoring

State Poller analyzes interface group membership before node groups. For each of the interface groups you created, as well as any of the preexisting ones you want to use, open the **Monitoring Configuration** dialog and the **Interface Settings** tab to create a custom set of instructions for how State Poller should handle that group. Your instructions will include:

- Enabling or disabling fault polling
- Setting the fault polling interval
- Enabling or disabling performance polling if you have the NNM iSPI for Performance
- Setting the performance polling interval if you have the NNM iSPI for Performance
- Setting performance management thresholds if you have the NNM iSPI for Performance
- Selecting whether unconnected interfaces (or unconnected interfaces hosting IP addresses) in the group should be monitored

You can configure different settings for each interface group. Remember that the State Poller evaluates the list in order from the lowest ordering number to the highest ordering number.

Best practice

Double-check your order numbers, keeping in mind that an object that qualifies for multiple groups has settings applied from the group with the lowest order number.

Configure Node Monitoring

If an object does not qualify for any configured interface group, State Poller evaluates the object for membership in node groups. Settings are applied to the first node group match from the lowest ordering number to the highest ordering number.

For each node group, open the **Monitoring Configuration** form, and then, open the **Node Settings** tab. Create a custom set of instructions as to how State Poller should handle that group. Your instructions can include:

- Enabling or disabling fault polling
- Setting the fault polling interval
- Enabling or disabling performance polling if you have the NNM iSPI for Performance
- Setting the performance polling interval if you have the NNM iSPI for Performance
- Setting performance management thresholds if you have the NNM iSPI for Performance
- Selecting whether unconnected interfaces (or unconnected interfaces hosting IP addresses) in the group should be monitored

You may configure different settings for each node group.

Best practice

Double-check the order numbers, keeping in mind that an object that qualifies for multiple groups has settings applied from the group with the lowest order number.

Verify Default Settings

State Poller applies the settings from the **Default Settings** tab for any object that does not match a defined interface setting or node setting. Review the settings on this tab to ensure they match your environment at the default level. For example, you would rarely poll all unconnected interfaces as a default setting.

Be sure you **Save and Close** all **Monitoring Configuration** dialog boxes all the way back to the console for your changes to be implemented.

Evaluate State Polling

This section lists ways to evaluate the progress and success of the monitoring settings.

Verify the Configuration for Network Monitoring

You can determine the settings that NNMi uses for monitoring a given node or interface, and you can initiate a status poll of a node at any time.

Is the interface or node a member of the right group?

You can verify which interfaces or nodes belong to a group by selecting one of the following in the **Configuration** workspace:

- Node Groups
- Interface Groups

Follow the instructions in the help to show the members of the group. Keep in mind that an object can be a member of multiple groups, and that another group may have a lower ordering number.

Alternatively, you can see the full list of groups to which the object belongs by opening the object (interface or node) and clicking the **Groups** tab. This list is alphabetical by group name and does not reflect the ordering numbers that determine which settings are applied.

If the object is not a member of a group:

1. Retrieve the Device Profile for the node in the inventory view.
2. Review the attribute mapping for the Device Profile under **Configuration > Device Profiles**.
3. Review the attribute requirements for the node group definition.

If you have a mismatch, you can adjust the category derived in the Device Profile to force that type of device to qualify for your node group. You may need to do an **Actions > Configuration Poll** to update the attributes for the node so that it qualifies.

Which settings are being applied?

To check the monitoring configuration in effect for a specific node, interface, or address, select that object in the appropriate inventory view, and select **Actions > Monitoring Settings**. NNMi displays the current monitoring settings.

Examine the values for **Fault Polling Enabled** and **Fault Polling Interval**. If these values are not as expected, look at the value for **Node Group** or **Interface Group** to see which ordered group match applied.

You may also need to check **Actions** > **Communication Settings** for the object to ensure traffic has not been disable for it.

Which data is being collected?

You can initiate a status poll of a specific device to validate that the expected types of polls (SNMP, ICMP) are being performed for that device. Select a node, and then click **Actions > Status Poll**. NNMi performs a real-time status check of the device. The output shows the types and results of the polls being performed. If the types of polls are not what you expect, check the monitoring settings for the node and the respective global, interface, or node settings of the monitoring configuration.

Evaluate the Performance of Status Polling

Evaluate the performance of status polling in your environment by using the information in the state poller health check to quantify and assess the operation of the state poller service.

Is the State Poller keeping up?

At any time, you can check the current health statistics about the state poller service with **Help > About Network Node Manager i-series**. Table 2 lists the state poller health information that NNMi displays in the **About HP Network Node Manager i-series** window.

Table 2 State Poller Health Information

Information	Description
Status	Overall status of the state poller service
Poll counters	• Collections requested in last minute • Collections completed in last minute • Collections in process
Time to execute skips in last minute	The number of regularly scheduled polls that did not complete within the configured polling interval. A non-zero value indicates that the polling engine is not keeping up or that targets are being polled faster than they can respond. • What to watch for: If this value continues to increase, there are problems communicating with the target or NNMi is overloaded.
Stale collections	A stale collection is a collection that has not received a response from the polling engine for at least 10 minutes. A healthy system should never have any stale collections. • What to watch for: If this value increases consistently, there is a problem with the polling engine.
Poller result queue length	• What to watch for: This value should be close to 0 most of the time. • Action to take: If this queue size is very large, ovjboss might be running out of memory.
State mapper queue length	• What to watch for: This value should be close to 0 most of the time. • Action to take: If this queue size is very large, then check the performance of the NNMi system and the NNMi database.

Tune State Polling

The performance of state polling is affected by the following key variables:

- The number of devices/interfaces to be polled
- The type of polling configured
- The frequency of polling each device

These variables are driven by your network management needs. If you are experiencing performance issues with status polling, consider the following configurations:

- Because polling settings for individual nodes are controlled through their membership in node groups and interface groups, make sure that the groups contain nodes or interfaces with similar polling requirements.
- If you are polling unconnected interfaces or interfaces that host IP addresses, check the configurations to make sure you are only polling the interfaces that are necessary. Enable these polls on the **Node Settings** or **Interface Settings** form (not as a global setting on the **Monitoring Configuration** form) to maintain the most specific control and to select the smallest subset of interfaces to poll.
- Remember that polling unconnected interfaces monitors *all* unconnected interfaces. To monitor only those unconnected interfaces that have IP addresses, enable polling of interfaces that host IP addresses.

Regardless of the monitoring configuration, status polling is dependent on network responsiveness and may be impacted by overall system performance. Although status polling with default polling intervals does not introduce much network load, if the performance of the network link between the server and the polled device is poor, status polling performance is poor. You can configure larger timeouts and a smaller number of retries to reduce the network load, but these configuration changes only go so far. Timely polling requires adequate network performance and sufficient system resources (CPU, memory).

Enabling or disabling the Component Health monitoring has no effect on timeliness of polling. It simply gathers additional MIB objects at the schedule time. However, disabling Component Health monitoring may reduce the amount of memory used by the State Poller.

Administration

This section contains the following chapters:

Enabling https for NNMi

By default, the HP Network Node Manager i Software web server uses the http protocol for transferring data to the NNMi console. This implementation is adequate for environments in which all users of the NNMi console are inside a secure corporate Intranet. If users outside of the Intranet need to access the NNMi console, you can enable http over secure socket layer (https) to encrypt the data that flows between the NNMi web server and the web browser.

For a more detailed description of secure sockets layer (SSL) technology and the configuration steps in this topic, go to:

http://tomcat.apache.org/tomcat-6.0-doc/ssl-howto.html

To enable the https protocol for the NNMi web server, complete the following steps:

- Obtain and Install a Public Key Certificate
- Update the server.xml File
- Communicate the New Connection Information to Users

Obtain and Install a Public Key Certificate

A public key certificate identifies the web server to the browser. This certificate can be self-signed or third-party.

Self-Signed Certificate

Any system administrator can create a self-signed certificate for a single web server. Self-signed certificates are useful when all users of the web server already have a trust relationship with the organization that provides the web application, in this case, the NNMi console.

NNMi includes a self-signed certificate in the `nnm.keystore` file and an empty trust store in the `nnm.truststore` file.

To install the NNMi-provided self-signed certificate, follow these steps:

1 Change to the directory that contains the NNMi trust store:

- *Windows*: `%NnmDataDir%\shared\nnm\certificates`
- *UNIX*: `$NnmDataDir/shared/nnm/certificates`

Run all of the commands in this procedure from the `certificates` directory.

2 Export the provided certificate to a certificate file:

a Run the following command:

— *Windows*:
```
%NnmInstallDir%\nonOV\jdk\b\bin\keytool -export \
-alias selfsigned -keystore nnm.keystore \
-file nnm.cert
```

— *UNIX*:
```
$NnmInstallDir/nonOV/jdk/b/bin/keytool -export \
-alias selfsigned -keystore nnm.keystore \
-file nnm.cert
```

b When prompted for the keystore password, enter: **nnmkeypass**

3 Import the certificate from the temporary file to the trust store:

a Run the following command:

— *Windows*:
```
%NnmInstallDir%\nonOV\jdk\b\bin\keytool -import \
-alias nnmi_https -trustcacerts \
-keystore nnm.truststore -file nnm.cert
```

— *UNIX*:
```
$NnmInstallDir/nonOV/jdk/b/bin/keytool -import \
-alias nnmi_https -trustcacerts \
-keystore nnm.truststore -file nnm.cert
```

b When prompted for the keystore password, enter: **ovpass**

c When prompted to trust the certificate, enter: **y**

Example output for importing the certificate into the trust store

The output from this command is of the form:

```
Owner: CN=NNMi_server.hp.com
Issuer: CN=NNMi_server.hp.com
Serial number: 494440748e5
Valid from: Tue Oct 28 10:16:21 MST 2008 until: Thu Oct 04
11:16:21 MDT 2108
Certificate fingerprints:
MD5:  29:02:D7:D7:D7:D7:29:02:29:02:29:02:29:02:29:02
SHA1: C4:03:7E:C4:03:7E:C4:03:7E:C4:03:7E:C4:03:7E:C4:03
Trust this certificate? [no]:  y
Certificate was added to keystore
```

4 Verify that the certificate was correctly imported into the trust store:

a Examine the contents of the key store:

— *Windows*:
%NnmInstallDir%\nonOV\jdk\b\bin\keytool -list \ -keystore nnm.keystore

— *UNIX*:
$NnmInstallDir/nonOV/jdk/b/bin/keytool -list \ -keystore nnm.keystore

When prompted for the keystore password, enter: **nnmkeypass**

Example key store output

The key store output is of the form:

```
Keystore type: jks
Keystore provider: SUN
Your keystore contains 1 entry
selfsigned, Oct 28, 2008, keyEntry,
Certificate fingerprint (MD5):
29:02:D7:D7:D7:D7:29:02:29:02:29:02:29:02:29:02
```

b Examine the contents of the trust store:

— *Windows*:
%NnmInstallDir%\nonOV\jdk\b\bin\keytool -list \ -keystore nnm.truststore

— *UNIX*:
$NnmInstallDir/nonOV/jdk/b/bin/keytool -list \ -keystore nnm.truststore

When prompted for the keystore password, enter: **ovpass**

Example trust store output

The trust store output is of the form:

```
Keystore type: jks
Keystore provider: SUN
Your keystore contains 1 entry
nnmi_https, Nov 14, 2008, trustedCertEntry,
Certificate fingerprint (MD5):
29:02:D7:D7:D7:D7:29:02:29:02:29:02:29:02:29:02
```

The trust store can include multiple certificates.

c Compare the certificate fingerprint from the key store output with the certificate fingerprints in the trust store output. If you do not find a match, repeat this procedure.

For more information about the `keytool` command, search for "Key and Certificate Management Tool" at **http://java.sun.com/**.

Update the server.xml File

The version of jboss that is delivered with NNMi 8.11 includes the Tomcat server version 6.0. The https configuration process for Tomcat 6.0 is different from the process for previous versions of Tomcat.

The Tomcat server is configured in the following file:

```
$jboss.home.dir/server/nms/deploy/jboss-web.deployer/server.xml
```

The default value of `$jboss.home.dir` is as follows:

- *Windows*: `%NnmInstallDir%\nonOV\jboss\nms`
- *UNIX*: `$NnmInstallDir/nonOV/jboss/nms`

Each connector block in the `server.xml` file specifies the parameters for a type of connection between the web server and the web browser. The `server.xml` file can contain multiple connector blocks.

To specify the https protocol, follow these steps:

1. Open the `server.xml` file in any text editor.

   For examples that show what you need to change in the `server.xml` file, see Example Connector Blocks.

2. Examine the connector block for the non-secure connection, which is in the `<!-- HTTP CONFIG -->` section of the file.

   - Leave the port definition as configured. For more information, see jboss Ports on page 81.
   - If you want users inside the Intranet to connect to the NNMi console with the http protocol, leave this default block in place.

     Other programs that integrate with NNMi use the http protocol for communicating with NNMi. Do not delete the http connector block.

   - If you want all users to use the https protocol for connecting to the NNMi console, change the address line in the http connector block to the local host address to enable local NNMi management communications:

     ```
     address=127.0.0.1
     ```

3. Uncomment the appropriate https connector block.

   For a self-signed certificate, use the first connector block in the `<!-- HTTPS/SSL CONFIG -->` section of the `server.xml` file.

4. Modify the https connector block as appropriate.

   - Leave the port definition as configured. For more information, see jboss Ports on page 81.
   - For the `keystoreFile` and `truststoreFile` parameters, supply the correct path to the certificate files.

   For information about the parameters in the connector block, refer to the Tomcat configuration help that is referenced earlier in this chapter.

5. Save the `server.xml` file.

6 Restart `ovjboss`:

```
ovstop ovjboss
ovstart ovjboss
```

jboss Ports

The `jboss.http.port` and `jboss.https.port` variables are used in the server configuration connector blocks to identify the port for the http or https protocol, respectively. These variables defined in the following file:

- *Windows*: `%NnmDataDir%\shared\nnm\conf\nnm.ports.properties`
- *UNIX*: `$NnmDataDir/shared/nnm/conf/nnm.ports.properties`

Several processes trust the `jboss.http.port` and `jboss.https.port` variables as the definitive port values for connecting to NNMi. If you need to change a port for connecting to NNMi, edit the corresponding variable value in the `nnm.ports.properties` file, and then restart `ovjboss`.

To identify the ports that are already in use on the computer, use the `netstat -a` command.

Example Connector Blocks

In the following examples, the bolded lines indicate commonly changed parameters.

Example connector block for a non-secure connection

The default connector block for the non-secure connection is in the `<!-- HTTP CONFIG -->` section of the `server.xml` file and is similar to the following example:

```
<Connector port="${jboss.http.port}"
    address="${jboss.bind.address}"
    maxThreads="250"
    maxHttpHeaderSize="8192"
    emptySessionPath="true"
    protocol="HTTP/1.1"
    enableLookups="false"
    redirectPort="${jboss.https.port:8443}"
    acceptCount="100"
    connectionTimeout="20000"
    disableUploadTimeout="true"
    URIEncoding="UTF-8" />
```

Example connector block for a secure connection with a self-signed certificate

The first connector block in the `<!-- HTTPS/SSL CONFIG -->` section of the `server.xml` file is for a self-signed certificate and is similar to the following example:

```
<Connector port="${jboss.https.port}"
    address="${jboss.bind.address}"
    maxThreads="250"
    acceptCount="100"
    connectionTimeout="20000"
    strategy="ms"
    maxHttpHeaderSize="8192"
    emptySessionPath="true"
    scheme="https"
    secure="true"
    SSLEnabled="true"
```

```
    keystoreFile="C:/Documents and Settings/All Users/
Application Data/HP/HP BTO Software/shared/nnm/certificates/
nnm.keystore"
    keystorePass="nnmkeypass"
    keyAlias="selfsigned"
    truststoreFile="C:/Documents and Settings/All Users/
Application Data/HP/HP BTO Software/shared/nnm/certificates/
nnm.truststore"
    truststorePass="ovpass"
    trustAlias="nnmi_https"
    clientAuth="false"
    sslProtocol="TLS"
    URIEncoding="UTF-8" />
```

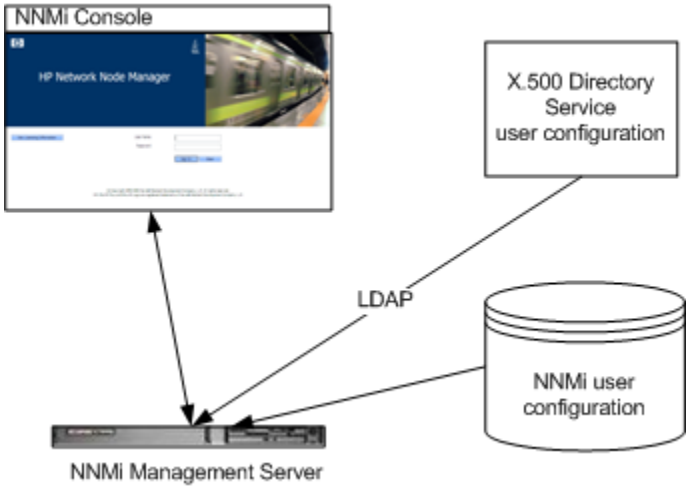
Communicate the New Connection Information to Users

For users outside of the Intranet, the new URL for accessing the NNMi console is as follows:

https://*<host_name>*:*<https_port>*/nnm/

If you are using the default https port (443), users do not need to include the port number in the URL.

Integrating NNMi with a Directory Service through LDAP



By default, sign-in access to the NNMi console is controlled through HP Network Node Manager i Software user accounts that are mapped to NNMi user roles. If your organization maintains user names and passwords in an X.500 directory service, you can configure NNMi to access the directory service during user signin to the NNMi console. This approach reduces the number of user names and passwords that NNMi operators must remember. It also reduces the maintenance burden on the NNMi administrator by eliminating the effort to change NNMi user passwords periodically and to delete user accounts when network operations personnel leave the department.

NNMi uses the lightweight directory access protocol (LDAP) for communicating with the directory service.

This chapter contains the following topics:

NNMi User Access Information and Configuration Options

Each NNMi user is defined by the combination of the following values:

- The **user name** uniquely identifies the NNMi user. The user name provides access to NNMi and receives incident assignments.
- The **password** is associated with the user name to control access to the NNMi console or NNMi command.
- The **NNMi role** assignment controls the information available and the type of actions that a user can take in the NNMi console. The role assignment also controls the availability of NNMi commands to the user.

As of version 8.11, NNMi provides several options for where the NNMi user access information is stored, as described in the following topics. Table 3 indicates the databases that store the NNMi user access information for each configuration option.

Table 3 Options for Storing User Information

Option	User Name	Password	Role Mapping
1	NNMi	NNMi	NNMi
2	Both	Directory Service	NNMi
3	Directory Service	Directory Service	Directory Service

When NNMi is integrated with a directory service for some or all of the user access information, the **About Network Node Manager i-series** window indicates the type of information that was obtained through LDAP queries.

Single sign-on between NNMi and other applications is not dependent on how the NNMi user access information is configured or where this information is stored.

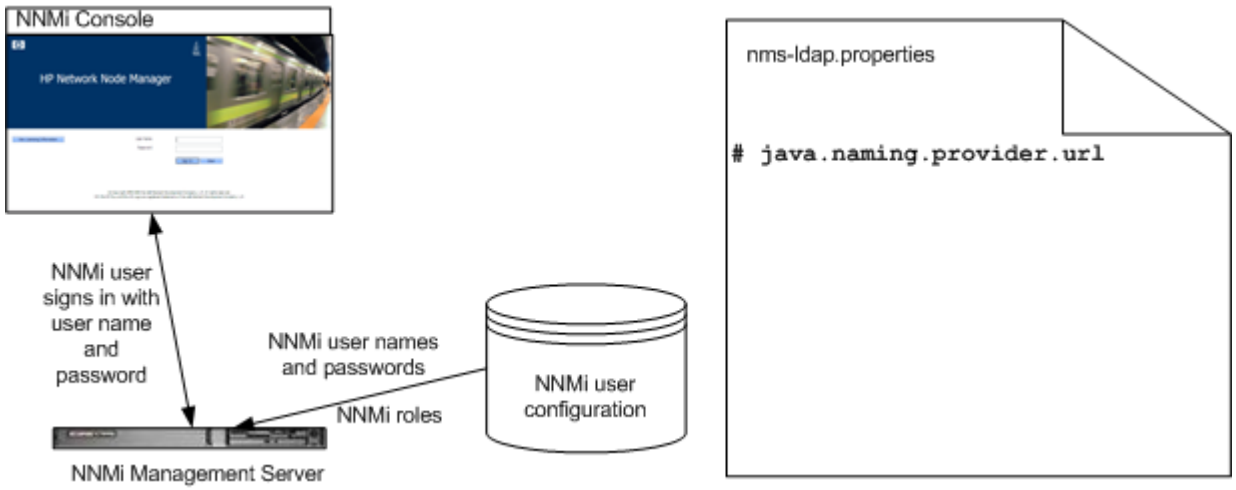
Option 1: All NNMi User Information in the NNMi Database

NNMi accesses the NNMi database for all user access information, which the NNMi administrator defines and maintains in the NNMi console. The user access information is local to NNMi. NNMi does not access a directory service, and the `nms-ldap.properties` file is ignored (as indicated by the commented line in Figure 2).

Figure 2 shows the information flow for this option, which is good for environments with a small number of NNMi users and for environments without a directory service.

For information about setting up all user information in the NNMi database, see *Control Access with NNMi Accounts* in the NNMi help. You do not need to read this chapter.

Figure 2 NNMi User Sign-in Information Flow for Option 1



Option 2: Some NNMi User Information in the NNMi Database and Some User Information in the Directory Service

NNMi accesses a directory service for the user name and password, which are defined externally to NNMi and are also available to other applications. The mapping of users to NNMi roles is maintained in the NNMi console. The configuration and maintenance of NNMi user access information is a joint effort as described here:

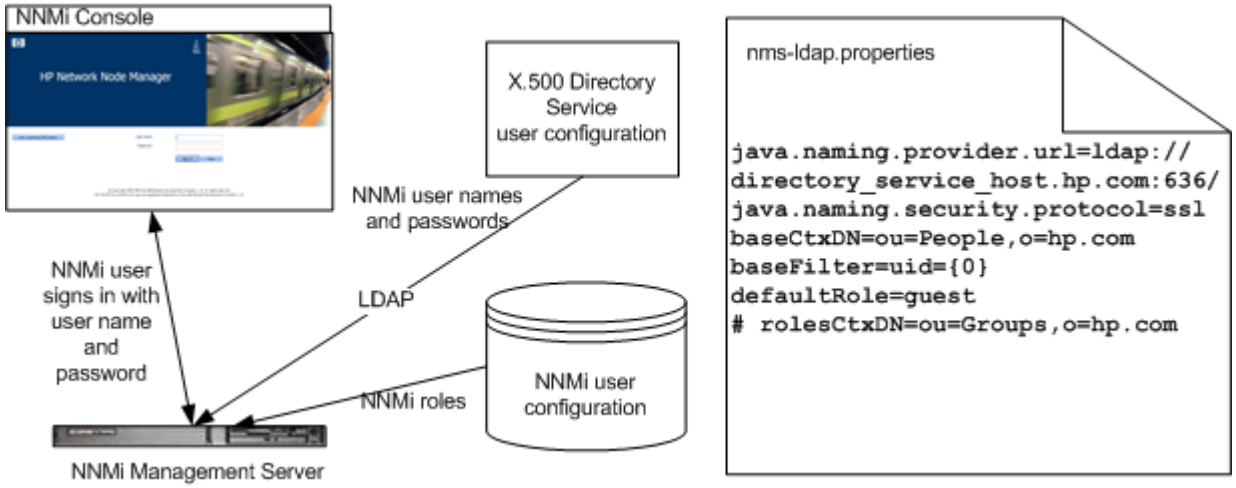
- The directory service administrator maintains the user names and password in the directory service.
- The NNMi administrator enters the user names (as defined in the directory service) and the NNMi role mappings in the NNMi console.
- The NNMi administrator configures the NNMi `nms-ldap.properties` file to describe the directory service database schema for user names to NNMi. (In Figure 3, the commented line indicates that NNMi does not pull NNMi role information from the directory service.)

Because user names must be entered in two places, user name maintenance must be performed in both places.

Figure 3 shows the information flow for this option, which is good for environments with a directory service and a small number of NNMi users, or environments where the NNMi administrator wants to control the user roles instead of requiring a directory service change for each role change. It is also good for environments where the directory service group definitions are not easily expandable.

For information about integrating with a directory service for the user name and password, see the rest of this chapter and *Control Access Using Both Directory Service and NNMi* in the NNMi help.

Figure 3 NNMi User Sign-in Information Flow for Option 2



Option 3: All NNMi User Information in the Directory Service

NNMi accesses a directory service for all user access information, which is defined externally to NNMi and is available to other applications. Membership in one or more directory service groups determines the NNMi role for the user. (A user with membership in multiple groups that define NNMi roles receives the most powerful of the roles.)

The directory service administrator maintains the user and group information in the directory service. The directory service administrator configures groups in the directory service, one group for each NNMi role.

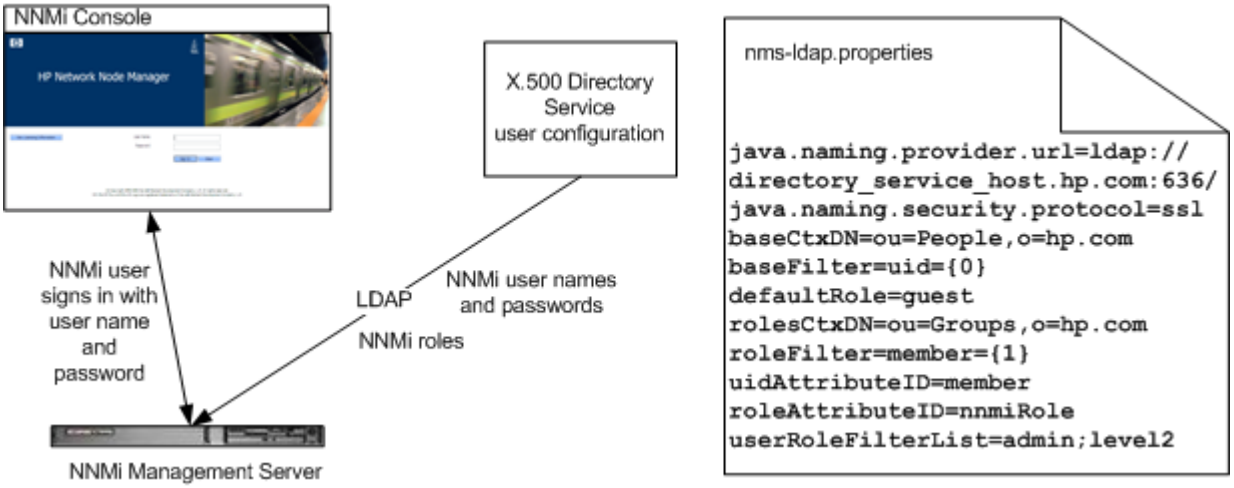
The NNMi administrator configures the NNMi `nms-ldap.properties` file to describe the directory service database schema for user names, user groups, and roles to NNMi.

Figure 4 shows the information flow for this option, which is good for environments where the directory service can be modified to include user groups that align with the people who need access to NNMi. Regardless of current configuration, this option requires modifications to the directory service group definitions.

Because this option is an expansion of the option 2 scenario, HP recommends that you first configure and verify NNMi user name and password retrieval from the directory service before beginning to configure NNMi role retrieval from the directory service.

For information about integrating with a directory service for all user information, see the rest of this chapter and *Control Access with a Directory Service* in the NNMi help.

Figure 4 NNMi User Sign-in Information Flow for Option 3



Configuring NNMi to Access a Directory Service

Directory service access is configured in the following file:

- *Windows*: `%NnmInstallDir%\nonOV\jboss\nms\server\nms\conf\props\nms-ldap.properties`
- *UNIX*: `$NnmInstallDir/nonOV/jboss/nms/server/nms/conf/props/nms-ldap.properties`

For detailed information about this file, see nms-ldap.properties Configuration File Reference on page 105. Also see Examples on page 109.

For detailed information about the general structure of a directory service, see Directory Service Queries on page 93.

For configuration option 2, complete the following tasks:

- Task 1: Back up the Current NNMi User Information
- Task 2: Optional. Configure Secure Communications to the Directory Service
- Task 3: Configure NNMi User Access from the Directory Service
- Task 4: Test the User Name and Password Configuration
- Task 8: Clean up to Prevent Unexpected Access to NNMi

For configuration option 3, complete the following tasks:

- Task 1: Back up the Current NNMi User Information
- Task 2: Optional. Configure Secure Communications to the Directory Service
- Task 3: Configure NNMi User Access from the Directory Service
- Task 4: Test the User Name and Password Configuration
- Task 5: Configure NNMi Role Retrieval from the Directory Service

If you plan to store NNMi roles in the directory service, the directory service must be configured with the NNMi roles. For more information, see Directory Service Configuration for Storing NNMi Roles on page 104.

- Task 6: Test the NNMi Role Configuration
- Task 7: Configure NNMi Roles for Incident Assignment
- Task 8: Clean up to Prevent Unexpected Access to NNMi

Task 1: Back up the Current NNMi User Information

Back up the user information in the NNMi database:

- *Windows*: **`%NnmInstallDir%\bin\nnmconfigexport.ovpl -c account \ -u <user> -p <password> -f NNMi_database_accounts.xml`**
- *UNIX*: **`$NnmInstallDir/bin/nnmconfigexport.ovpl -c account \ -u <user> -p <password> -f NNMi_database_accounts.xml`**

Task 2: Optional. Configure Secure Communications to the Directory Service

If the directory service requires the use of secure sockets layer (SSL), import your company's certificate into the NNMi trust store as described in Configuring an SSL Connection to the Directory Service on page 92.

Task 3: Configure NNMi User Access from the Directory Service

Complete this task for configuration options 2 and 3. Follow the appropriate procedure for your directory service. This task includes the following sections:

- Simple Approach for Microsoft Active Directory
- Simple Approach for Other Directory Services

Simple Approach for Microsoft Active Directory

1. If you have not already done so, install the latest consolidated NNMi patch.

   For LDAP access to Active Directory, NNMi must be running at least version 8.1x patch 3.

2. Back up the `nms-ldap.properties` file that was shipped with NNMi, and then open the file in any text editor.
3. Overwrite the file contents with the following text:

```
java.naming.provider.url=ldap://<myldapserver>:389/

bindDN=<mydomain>\\<myusername>
bindCredential=<mypassword>

baseCtxDN=CN=Users,DC=<myhostname>,DC=<mycompanyname>,DC=<mysuffix>
baseFilter=CN={0}

defaultRole=guest

#rolesCtxDN=CN=Users,DC=<myhostname>,DC=<mycompanyname>,DC=<mysuffix>
roleFilter=member={1}
uidAttributeID=member
roleAttributeIsDN=true
roleNameAttributeID=info
roleAttributeID=memberOf
userRoleFilterList=admin;level2;level1
```

4. In the following line:

   ```
   java.naming.provider.url=ldap://<myldapserver>:389/
   ```

   Replace *<myldapserver>* with the fully-qualified hostname of the Active Directory server (for example: `myserver.example.com`).

5. In the following lines:

   ```
   bindDN=<mydomain>\\<myusername>
   bindCredential=<mypassword>
   ```

   Make the following substitutions:

   - Replace *<mydomain>* with the name of Active Directory domain.
   - Replace *<myusername>* and *<mypassword>* with a user name and password for accessing the Active Directory server. Because the password is stored in plain text, specify a user name with read-only access to the directory service.

6 In the following line:

```
baseCtxDN=CN=Users,DC=<myhostname>,DC=<mycompanyname>,
  DC=<mysuffix>
```

Replace *<myhostname>*, *<mycompanyname>*, and *<mysuffix>* with the components of the fully-qualified hostname of the Active Directory server (for example, for the hostname `myserver.example.com`, specify: `DC=myserver,DC=example,DC=com`).

Simple Approach for Other Directory Services

1 Back up the `nms-ldap.properties` file that was shipped with NNMi, and then open the file in any text editor.

2 In the following line:

```
#java.naming.provider.url=ldap://<myldapserver>:389/
```

Do the following:

- Uncomment the line (by deleting the `#` character).
- Replace *<myldapserver>* with the fully-qualified hostname of the directory server (for example: `myserver.example.com`).

3 In the following line:

```
baseCtxDN=ou=People,o=myco.com
```

Replace `ou=People,o=myco.com` with the portion of the directory service domain that stores user records.

4 In the following line:

```
baseFilter=uid={0}
```

Replace `uid` with the user name attribute from the directory service domain.

Task 4: Test the User Name and Password Configuration

1 In the `nms-ldap.properties` file, set `defaultRole=guest` for testing purposes. (You can change this value at any time.)

2 Save the `nms-ldap.properties` file.

3 Sign in to the NNMi console with a user name and password that are defined in the directory service.

Run this test with a user name that is not already defined in the NNMi database.

4 Verify the user name and role (guest) in the title bar of the NNMi console.

If user signin does not work correctly, check the configuration, and then repeat step 1 through step 4 until you see the expected result.

a Verify that you completed Task 3 on page 88 correctly.

b Follow the detailed configuration process in User Identification on page 99.

After each test, sign out of the NNMi console to clear the session credentials.

5 After you can sign in, choose your strategy:

- If you plan to store role assignments in the NNMi database (configuration option 2), continue with Task 8 on page 91.
- If you plan to store role assignments in the directory service (configuration option 3), continue with Task 5.

Task 5: Configure NNMi Role Retrieval from the Directory Service

Complete this task for configuration option 3. Follow the appropriate procedure for your directory service. This task includes the following sections:

- Simple Approach for Microsoft Active Directory
- Simple Approach for Other Directory Services

Simple Approach for Microsoft Active Directory

1 Open the `nms-ldap.properties` file in any text editor.

2 In the following line:

```
#rolesCtxDN=CN=Users,DC=<myhostname>,DC=<mycompanyname>,
  DC=<mysuffix>
```

Do the following:

- Uncomment the line (by deleting the `#` character).
- Replace *<myhostname>*, *<mycompanyname>*, and *<mysuffix>* with the components of the fully-qualified hostname of the Active Directory server (for example, for the hostname `myserver.example.com`, specify: `DC=myserver,DC=example,DC=com`).

Simple Approach for Other Directory Services

1 Open the `nms-ldap.properties` file in any text editor.

2 In the following line:

```
#rolesCtxDN=ou=Groups,o=myco.com
```

Do the following:

- Uncomment the line (by deleting the `#` character).
- Replace `ou=Groups,o=myco.com` with the portion of the directory service domain that stores group records.

3 In the following line:

```
roleFilter=member={1}
```

Replace `member` with the name of the group attribute that stores the directory service user ID in the directory service domain.

Task 6: Test the NNMi Role Configuration

1 Save the `nms-ldap.properties` file.

2 Sign in to the NNMi console with a user name and password that are defined in the directory service.

Run this test with a user name that is not already defined in the NNMi database.

3 Verify the user name and role (as configured in the directory service) in the title bar of the NNMi console.

If user signin does not work correctly, check the configuration, and then repeat step 1 through step 3 until you see the expected result.

a Verify that you completed Task 5 on page 90 correctly.

b Follow the detailed configuration process in Role Identification on page 101.

After each test, sign out of the NNMi console to clear the session credentials.

Task 7: Configure NNMi Roles for Incident Assignment

1 Open the `nms-ldap.properties` file in any text editor.

2 Modify the `userRoleFilterList` parameter value to specify the NNMi roles to which NNMi operators can assign incidents.

3 Save the `nms-ldap.properties` file.

4 Sign in to the NNMi console with a user name and password that are defined in the directory service.

5 In any incident view, select an incident, and then click **Actions > Assign Incident**. Verify that you can assign the incident to a user in each of the roles specified by the `userRoleFilterList` parameter.

6 Repeat step 1 through step 5 until you can assign an incident to each configured role type.

Task 8: Clean up to Prevent Unexpected Access to NNMi

1 Optional. Change the value of, or comment out, the `defaultRole` parameter in the `nms-ldap.properties` file.

2 Reset the user access information in the NNMi database:

- To store role assignments in the NNMi database (configuration option 2), do the following in the NNMi console:
  - Remove any pre-existing user access information.
  - Create role assignments for the directory service user names.
  - Update incident ownership.

  For detailed instructions, see *Control Access Using Both Directory Service and NNMi* in the NNMi help.
- If you configured role assignment from the directory service (configuration option 3), do the following in the NNMi console:
  - Remove any pre-existing user access information.
  - Update incident ownership.

  For detailed instructions, see *Control Access with a Directory Service* in the NNMi help.

Configuring an SSL Connection to the Directory Service

By default, when you enable directory service communications, NNMi uses the LDAP protocol for retrieving data from a directory service. If your directory service requires an SSL connection, you must enable the SSL protocol to encrypt the data that flows between NNMi and the directory service.

SSL requires a trust relationship between the directory service host and the NNMi management server. To enable this trust relationship, add a certificate to the NNMi trust store. The certificate confirms the identity of the directory service host to the NNMi management server.

To install a trust store certificate for SSL communications, follow these steps:

1. Obtain your company's trust store certificate from the directory server. The directory service administrator should be able to give you a copy of this text file.
2. Change to the directory that contains the NNMi trust store:
   - *Windows*: `%NnmDataDir%\shared\nnm\certificates`
   - *UNIX*: `$NnmDataDir/shared/nnm/certificates`

   Run all of the commands in this procedure from the `certificates` directory.
3. Import your company's trust store certificate into the NNMi trust store:
   a. Run the following command:
      — *Windows*:
      **%NnmInstallDir%\nonOV\jdk\b\bin\keytool -import **
      **-alias nnmi_ldap -keystore nnm.truststore **
      -file *<Directory_Server_Certificate.txt>*
      — *UNIX*:
      **$NnmInstallDir/nonOV/jdk/b/bin/keytool -import **
      **-alias nnmi_ldap -keystore nnm.truststore **
      -file *<Directory_Server_Certificate.txt>*

      Where *<Directory_Server_Certificate.txt>* is your company's trust store certificate.
   b. When prompted for the keystore password, enter: **ovpass**
   c. When prompted to trust the certificate, enter: **y**

      The output from this command is of the form:

Example output for importing a certificate into the trust store

```
Owner: CN=NNMi_server.example.com
Issuer: CN=NNMi_server.example.com
Serial number: 494440748e5
Valid from: Tue Oct 28 10:16:21 MST 2008 until: Thu Oct 04
11:16:21 MDT 2108
Certificate fingerprints:
MD5:  29:02:D7:D7:D7:D7:29:02:29:02:29:02:29:02:29:02
SHA1: C4:03:7E:C4:03:7E:C4:03:7E:C4:03:7E:C4:03:7E:C4:03
Trust this certificate? [no]:  y
Certificate was added to keystore
```

4 Examine the contents of the trust store:

- *Windows*:
  **%NnmInstallDir%\nonOV\jdk\b\bin\keytool -list **
  -keystore nnm.truststore
- *UNIX*:
  **$NnmInstallDir/nonOV/jdk/b/bin/keytool -list **
  -keystore nnm.truststore

When prompted for the keystore password, enter: **ovpass**

Example trust store output

The trust store output is of the form:

```
Keystore type: jks
Keystore provider: SUN
Your keystore contains 1 entry
nnmi_ldap, Nov 14, 2008, trustedCertEntry,
Certificate fingerprint (MD5):
29:02:D7:D7:D7:D7:29:02:29:02:29:02:29:02:29:02
```

The trust store can include multiple certificates.

5 Restart `ovjboss`:

```
ovstop ovjboss
ovstart ovjboss
```

For more information about the `keytool` command, search for "Key and Certificate Management Tool" at **http://java.sun.com/**.

Directory Service Queries

NNMi uses LDAP to communicate with a directory service. NNMi sends a request, and the directory service returns stored information. NNMi cannot alter the information that is stored in the directory service.

This section contains the following topics:

Directory Service Access

LDAP queries to a directory service use the following format:

ldap://<*directory_service_host*>:<*port*>/<*search_string*>

- `ldap` is the protocol indicator. Use this indicator for both standard connections and SSL connections to the directory service.
- <*directory_service_host*> is the fully-qualified name of the computer that hosts the directory service.
- <*port*> is the port that the directory service uses for LDAP communication. The default port for non-SSL connections is 389. The default port for SSL connections is 636.
- <*search_string*> contains the information request. For more information, see Directory Service Content and RFC 1959, *An LDAP URL Format*, which is available at:
  http://labs.apache.org/webarch/uri/rfc/rfc1959.txt

You can enter an LDAP query as a URL in a web browser to verify that you have the correct access information and the correct structure for the search string.

If the directory service (for example, Active Directory) does not allow anonymous access, the directory service denies LDAP queries from a web browser. In this case, you can use a commercially available LDAP browser to validate your configuration parameters.

Directory Service Content

A directory service stores information such as user names, passwords, and group membership. To access the information in a directory service, you must know the distinguished name that references the storage location of the information. For sign-in applications, the distinguished name is a combination of variable information (such as a user name) and fixed information (such as the storage location of user names). The elements that make up a distinguished name depend on the structure and content of a given directory service.

The following examples show possible definitions for a group of users called USERS-NNMi-Admin. This group lists the directory service user IDs that have administrative access to NNMi. The following information pertains to these examples:

- The Active Directory example is for the Windows operating system.
- The other directory services example is for UNIX operating systems.
- The file shown in each example is a portion of a lightweight directory interchange format (LDIF) file. LDIF files provide for sharing directory service information.
- Each example also includes a figure that shows a graphical representation of the directory service domain, which provides an expanded view of the information in the LDIF file excerpt.

Example content structure for Active Directory

In this example, the following items are of interest:

- The distinguished name of the user John Doe is:
  `CN=john.doe@example.com,OU=Users,OU=Accounts,DC=example,DC=com`
- The distinguished name of the group USERS-NNMi-Admin is:
  `CN=USERS-NNMi-Admin,OU=Groups,OU=Accounts,DC=example,DC=com`
- The group attribute that stores the directory service user ID is: `member`
- The group attribute that stores the NNMi role is: `info`

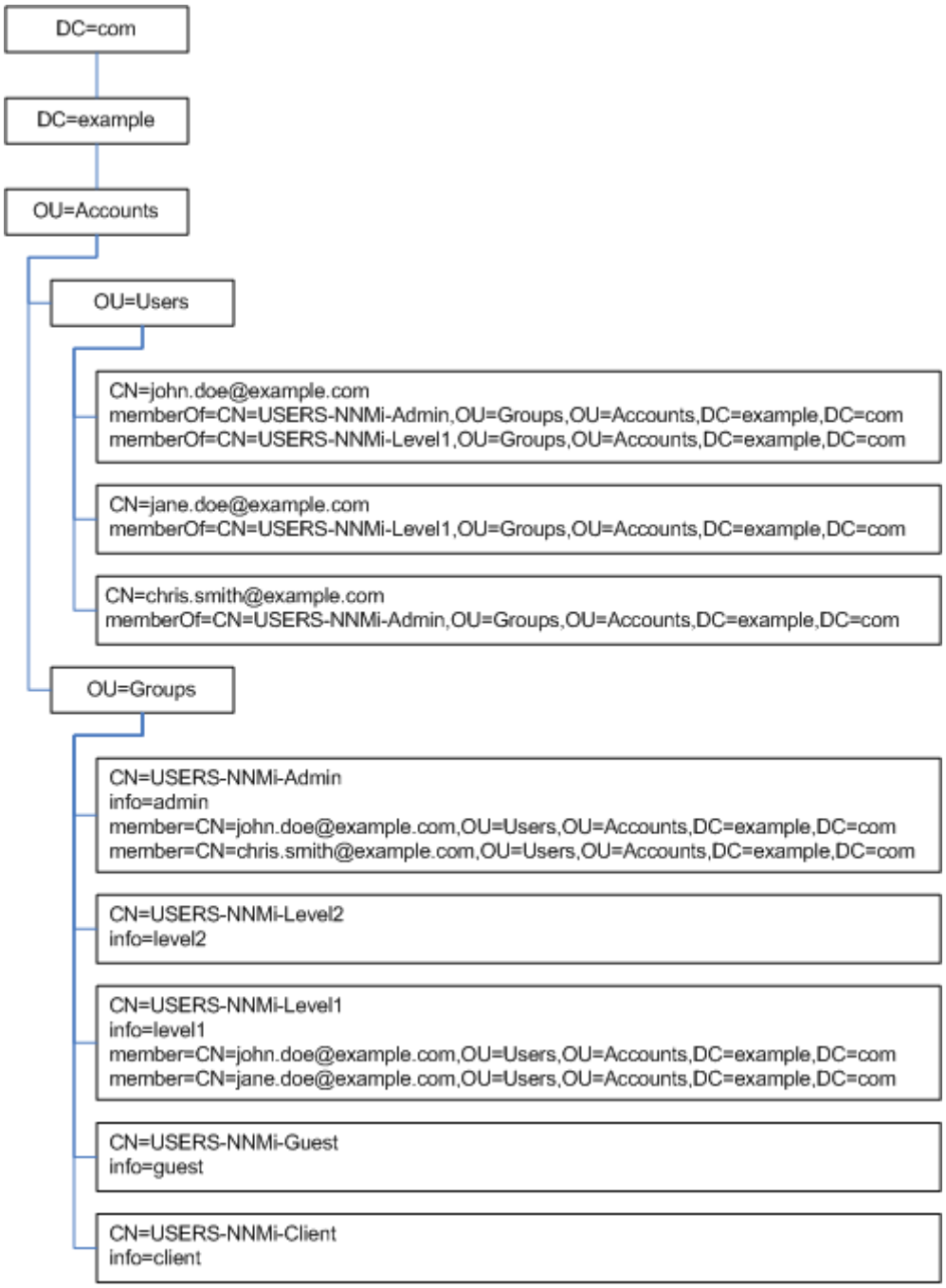
  The `info` attribute has been repurposed to store the NNMi role.

Example LDIF file excerpt:

```
groups |USERS-NNMi-Admin
dn: CN=USERS-NNMi-Admin,OU=Groups,OU=Accounts,DC=example,DC=com
info: admin
cn: USERS-NNMi-Admin
description: Group of users for NNMi administration.
member: CN=john.doe@example.com,OU=Users,OU=Accounts,
  DC=example,DC=com
member: CN=chris.smith@example.com,OU=Users,OU=Accounts,
  DC=example,DC=com
```

Figure 5 on page 96 illustrates this directory service domain.

Figure 5 Example Domain for Active Directory



Example content structure for other directory services

In this example, the following items are of interest:

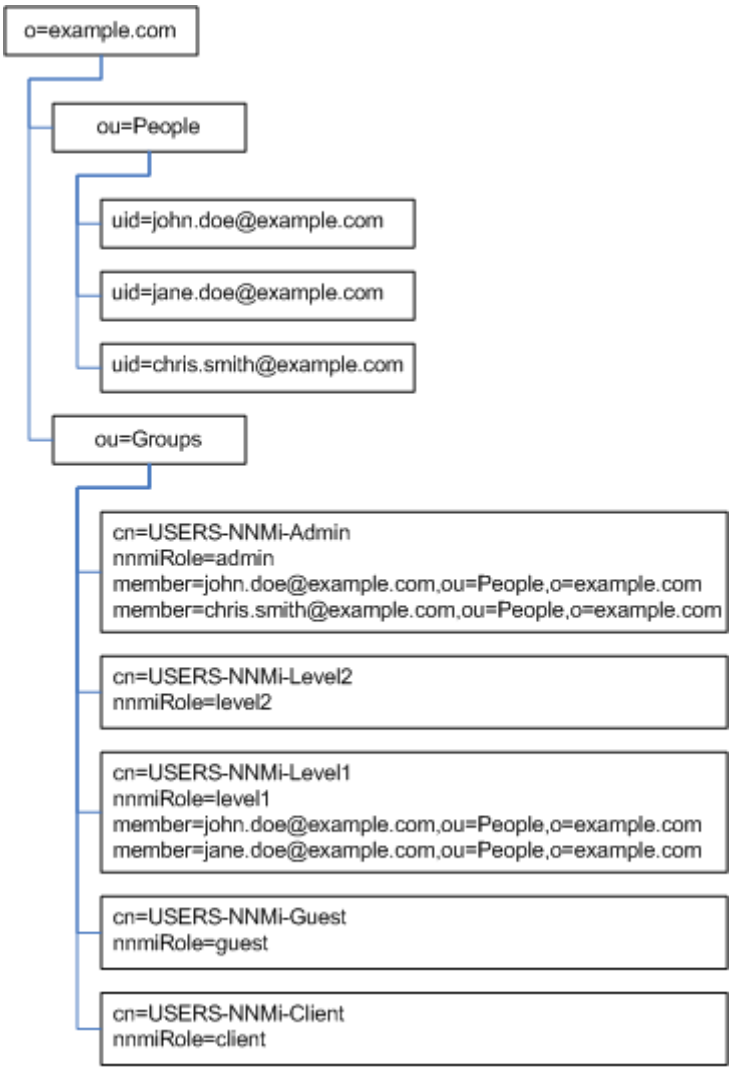
- The distinguished name of the user John Doe is: `uid=john.doe@example.com,ou=People,o=example.com`
- The distinguished name of the group USERS-NNMi-Admin is: `cn=USERS-NNMi-Admin,ou=Groups,o=example.com`
- The group attribute that stores the directory service user ID is: `member`
- The group attribute that stores the NNMi role is: `nnmiRole`

  The `nnmiRole` attribute was created specifically for this purpose.

Example LDIF file excerpt:

```
groups |USERS-NNMi-Admin
dn: cn=USERS-NNMi-Admin,ou=Groups,o=example.com
nnmiRole: admin
cn: USERS-NNMi-Admin
description: Group of users for NNMi administration.
member: uid=john.doe@example.com,ou=People,o=example.com
member: uid=chris.smith@example.com,ou=People,o=example.com
```

Figure 6 Example Domain for Other Directory Services



Information Owned by the Directory Service Administrator

Table 4 and Table 5 list the information to obtain from the directory service administrator before configuring NNMi for LDAP access to a directory service.

- If you plan to use the directory service for user names and passwords only (configuration option 2), gather the information for Table 4.
- If you plan to use the directory service for all NNMi access information (configuration option 3), gather the information for Table 4 and Table 5.

Table 4 Information for Retrieving User Names and Passwords from a Directory Service

Information	Active Directory Example	Other Directory Services Example
The fully-qualified name of the computer that hosts the directory service	`directory_service_host.example.com`	
The port that the directory service uses for LDAP communication	• 389 for non-SSL connections • 636 for SSL connections	
Does the directory service require an SSL connection?	If yes, obtain a copy of your company's trust store certificate and see Configuring an SSL Connection to the Directory Service on page 92.	
The distinguished name for one user name that is stored in the directory service (to demonstrate the directory service domain)	`CN=john.doe@example.com, OU=Users,OU=Accounts, DC=example,DC=com`	`uid=john.doe@example.com, ou=People,o=example.com`

Table 5 Information for Retrieving NNMi Roles from a Directory Service

Information	Active Directory Example	Other Directory Services Example
The distinguished name for identifying the groups to which a user is assigned	The `memberOf` user attribute identifies the groups.	• `ou=Groups,o=example.com` • `cn=USERS-NNMi-*, ou=Groups,o=example.com`
The method of identifying a user within a group	• `CN=john.doe@example.com, OU=Users,OU=Accounts, DC=example,DC=com` • `CN=john.doe@example.com`	• `cn=john.doe@example.com, ou=People,o=example.com` • `cn=john.doe@example.com`
The group attribute that stores the directory service user ID	`member`	`member`

Table 5 Information for Retrieving NNMi Roles from a Directory Service (cont'd)

Information	Active Directory Example	Other Directory Services Example
The names of the groups that the directory service administrator has created for mapping users to NNMi roles	• CN=USERS-NNMi-Admin, OU=Groups,OU=Accounts, DC=example,DC=com • CN=USERS-NNMi-Level2, OU=Groups,OU=Accounts, DC=example,DC=com • CN=USERS-NNMi-Level1, OU=Groups,OU=Accounts, DC=example,DC=com • CN=USERS-NNMi-Client, OU=Groups,OU=Accounts, DC=example,DC=com • CN=USERS-NNMi-Guest, OU=Groups,OU=Accounts, DC=example,DC=com	• cn=USERS-NNMi-Admin, ou=Groups,o=example.com • cn=USERS-NNMi-Level2, ou=Groups,o=example.com • cn=USERS-NNMi-Level1, ou=Groups,o=example.com • cn=USERS-NNMi-Client, ou=Groups,o=example.com • cn=USERS-NNMi-Guest, ou=Groups,o=example.com
The name of the attribute in each of the new groups that stores the NNMi role	info	nnmiRole

User Identification

User identification applies to configuration options 2 and 3.

The distinguished name for user identification is the fully-qualified method of locating one user in the directory service. NNMi passes the user distinguished name in an LDAP request to the directory service.

In the nms-ldap.properties file, the user distinguished name is the concatenation of the baseFilter value and the baseCtxDN value. If the password returned by the directory service matches the sign-in password that the user entered into the NNMi console, then user signin continues.

For configuration option 2, NNMi examines the following information and grants the user the role with the more privileges:

- The value of the defaultRole parameter in the nms-ldap.properties file
- The role assigned to this user in the NNMi console

For configuration option 3, NNMi determines the user role as described in Role Identification on page 101.

Active Directory user identification example

If baseFilter is set to CN={0}, baseCtxDN is set to OU=Users,OU=Accounts,DC=example,DC=com, and a user signs in to NNMi as john.doe, then the string passed to the directory service is:

```
CN=john.doe,OU=Users,OU=Accounts,DC=example,DC=com
```

Other directory services user identification example

If baseFilter is set to uid={0}@example.com, baseCtxDN is set to ou=People,o=example.com, and a user signs in to NNMi as john.doe, then the string passed to the directory service is:

```
uid=john.doe@example.com,ou=People,o=example.com
```

Configuring NNMi User Access from the Directory Service (Detailed Approach)

If the simple approach described in Task 3 on page 88 did not work correctly, follow these steps:

1. Obtain the information listed in Table 4 on page 98 from the directory service administrator.
2. Complete the appropriate procedure:
   - *LDAP browser approach for Active Directory and other directory services*: See Determining How the Directory Service Identifies a User (LDAP Browser Approach) on page 100.
   - *Web browser approach for other directory services*: See Determining How the Directory Service Identifies a User (Web Browser Approach) on page 101.
3. Open the `nms-ldap.properties` file in any text editor.

   For detailed information about the `nms-ldap.properties` file, see nms-ldap.properties Configuration File Reference on page 105.
4. Set the `java.naming.provider.url` parameter to the URL for accessing the directory service through LDAP.
   - *LDAP browser approach*: Obtain this information from the LDAP browser configuration.
   - *Web browser approach*: Include the values of <*directory_service_host*> and <*port*> from Determining How the Directory Service Identifies a User (Web Browser Approach) on page 101.
5. If you configured secure communications to the directory service, uncomment (or add) the following line:

   ```
   java.naming.security.protocol=ssl
   ```
6. (Active Directory) Set the `bindDN` and `bindCredential` parameters as follows:
   - Replace *<mydomain>* with the name of Active Directory domain.
   - Replace *<myusername>* and *<mypassword>* with a user name and password for accessing the Active Directory server. Because the password is stored in plain text, specify a user name with read-only access to the directory service.
7. Set the `baseCtxDN` parameter to the elements of the distinguished user name that are the same for multiple users.
8. Set the `baseFilter` parameter to correlate user names as they are entered for NNMi signin to the way user names are stored in the directory service.

   This value is the element of the distinguished user name that changes for each user. Replace the actual user name with the expression `{0}`.
9. Test the configuration as described in Task 4 on page 89.

Determining How the Directory Service Identifies a User (LDAP Browser Approach)

In a third-party LDAP browser, do the following:

1. Navigate to the portion of the directory service domain that stores group information.
2. Identify a group of users, and then examine the format of the distinguished names for the users associated with that group.

Determining How the Directory Service Identifies a User (Web Browser Approach)

1 In a supported web browser, enter the following URL:

ldap://<*directory_service_host*>:<*port*>/<*user_search_string*>

- <*directory_service_host*> is the fully-qualified name of the computer that hosts the directory service.
- <*port*> is the port that the directory service uses for LDAP communication.
- <*user_search_string*> is the distinguished name for one user name that is stored in the directory service.

2 Evaluate the results of the directory service access test.

- If the request times out or you see a message that the directory service could not be reached, verify the values of <*directory_service_host*> and <*port*>, and then repeat step 1.
- If you see a message that the directory service does not contain the requested entry, verify the value of <*user_search_string*>, and then repeat step 1.
- If you see the appropriate user record, the access information is correct. The value of <*user_search_string*> is the distinguished user name.

Role Identification

Role identification applies to configuration option 3.

NNMi determines a user's NNMi role by examining the user's group membership within the directory service. In the directory service groups and the NNMi configuration files, short text strings identify the NNMi roles. Table 6 maps the text strings to the role names that appear in the NNMi console.

Table 6 NNMi Role Name Mappings

Role Name in the NNMi Console	Text String in Directory Service Groups and NNMi Configuration Files
Administrator	`admin`
Operator Level 2	`level2`
Operator Level 1	`level1`
Guest	`guest`
Web Service Client	`client`

Active Directory Role Identification

A user definition in the directory service domain includes the directory service groups to which the user belongs. Each group name is indicated by a separate entry of a certain user attribute, commonly called `memberOf`. In the `nms-ldap.properties` file, the `roleAttributeID` parameter specifies the name of the user attribute that stores the user's group membership.

For each listed group, NNMi determines whether that group definition includes an NNMi role. In the `nms-ldap.properties` file, the `roleNameAttributeID` parameter specifies the name of the group attribute that stores the text string for an NNMi role

(as defined in Table 6). NNMi retrieves the value of that attribute to determine what role to assign to the NNMi user. (When a user is assigned multiple roles, NNMi uses the role with more privileges.)

Other Directory Services Role Identification

A group definition in the directory service domain includes the directory service users who belong to that group.

In the `nms-ldap.properties` file, the `roleFilter` value specifies how group members are identified, and the `rolesCtxDN` value specifies where in the directory service domain the groups are defined. For example, if `roleFilter` is set to `member={1}`, `rolesCtxDN` is set to `ou=Groups,o=example.com`, and the authenticated user distinguished name is `john.doe@example.com`, then NNMi queries the directory services for all groups whose fully-qualified distinguished names end in `ou=Groups,o=example.com`.

NNMi parses the list of groups returned by the directory service to determine which groups include the specification:

```
member=uid=john.doe@example.com,ou=People,o=example.com
```

Finally, NNMi parses the list of groups with the appropriate `member` specification to determine if any group has the attribute that corresponds to an NNMi role.

In the `nms-ldap.properties` file, the value of `roleAttributeID` is the name of the group attribute that stores the text string for an NNMi role (as defined in Table 6). NNMi retrieves the value of that attribute to determine what role to assign to the NNMi user. (When a user is assigned multiple roles, NNMi uses the role with more privileges.)

In the `nms-ldap.properties` file, the value of `uidAttributeID` is the name of the group attribute that stores the user name. NNMi retrieves the value of that attribute to cross-check the configuration.

Configuring NNMi Role Retrieval from the Directory Service (Detailed Approach)

If the simple approach described in Task 5 on page 90 did not work correctly, follow these steps:

1. Obtain the information listed in Table 5 on page 98 from the directory service administrator.
2. Complete the appropriate procedure:
   - *LDAP browser approach for Active Directory*: See Determining How the Directory Service Identifies a Group and Group Membership (LDAP Browser Approach for Active Directory) on page 103.
   - *LDAP browser approach for other directory services*: See Determining How the Directory Service Identifies a Group and Group Membership (LDAP Browser Approach for Other Directory Services) on page 103.
   - *Web browser approach for other directory services*: See Determining How the Directory Service Identifies a Group (Web Browser Approach) on page 104.
3. Open the `nms-ldap.properties` file in any text editor.

For detailed information about the `nms-ldap.properties` file, see nms-ldap.properties Configuration File Reference on page 105.

4 Set the `rolesCtxDN` parameter to the elements of the distinguished group name that are the same for multiple groups.

5 Set the `roleFilter` parameter to correlate user names to the way user names are stored for groups in the directory service. Replace the actual user name with one of the following expressions:

- Use `{0}` to denote the user name entered for signin (for example, `john.doe`).
- Use `{1}` to denote the distinguished name of the authenticated user as returned by the directory service (for example, `uid=john.doe@example.com,ou=People,o=example.com`).

6 Set the `uidAttributeID` parameter to the name of the group attribute that stores the user ID.

7 Set the `roleAttributeID` parameter as follows:

- For Active Directory, specify the name of the *user* attribute that indicates the groups to which the user belongs.
- For other directory services, specify the name of the *group* attribute that stores the NNMi role text string (one of the values listed in the second column of Table 6 on page 101).

8 Set the `roleAttributeIsDN` parameter as follows:

- For Active Directory, set this parameter to `true`.
- For other directory services, set this parameter to `false`.

9 (Active Directory) Set the `roleNameAttributeID` parameter to the name of the *group* attribute that stores the NNMi role text string (one of the values listed in the second column of Table 6 on page 101).

10 Test the configuration as described in Task 6 on page 90.

Determining How the Directory Service Identifies a Group and Group Membership (LDAP Browser Approach for Active Directory)

In a third-party LDAP browser, do the following:

1 Navigate to the portion of the directory service domain that stores user information.

2 Identify a user who requires access to NNMi, and then examine the format of the distinguished names for the groups associated with that user.

3 Navigate to the portion of the directory service domain that stores group information.

4 Identify the groups that correspond to NNMi roles, and then examine the format of the names for the users associated with a group.

Determining How the Directory Service Identifies a Group and Group Membership (LDAP Browser Approach for Other Directory Services)

In a third-party LDAP browser, do the following:

1 Navigate to the portion of the directory service domain that stores group information.

2 Identify the groups that correspond to NNMi roles, and then examine the format of the distinguished names for those groups.

3 Also examine the format of the names for the users associated with a group.

Determining How the Directory Service Identifies a Group (Web Browser Approach)

1 In a supported web browser, enter the following URL:

ldap://<*directory_service_host*>:<*port*>/<*group_search_string*>

- <*directory_service_host*> is the fully-qualified name of the computer that hosts the directory service.
- <*port*> is the port that the directory service uses for LDAP communication.
- <*group_search_string*> is the distinguished name for a group name that is stored in the directory service, for example: `cn=USERS-NNMi-Admin,ou=Groups,o=example.com`

2 Evaluate the results of the directory service access test.

- If you see a message that the directory service does not contain the requested entry, verify the value of <*group_search_string*>, and then repeat step 1.
- If you see the appropriate list of groups, the access information is correct.

3 Examine the group properties to determine the format of the names for the users associated with that group.

Directory Service Configuration for Storing NNMi Roles

If you plan to store NNMi roles in the directory service (configuration option 3), the directory service must be configured with NNMi role information. Ideally, the directory service already contains appropriate user groups. If this is not the case, the directory service administrator can create new user groups specifically for NNMi role assignment. Either way, the directory service administrator must maintain a group attribute for the NNMi role. This new attribute corresponds to the `roleNameAttributeID` parameter (for Active Directory) or to the `roleAttributeID` parameter (for other directory services) in the `nms-ldap.properties` file. Within the directory service, the possible values for the role attribute are defined in Table 6 on page 101.

Because directory service configuration and maintenance procedures depend on the specific directory service software and your company's policies, those procedures are not documented here.

Troubleshooting the Directory Service Integration

Ensure that the directory service contains the records that you expect. Use a web browser or a third-party LDAP browser to verify that you have the correct information.

Information about the format of a query to a directory service can be found in RFC 1959, *An LDAP URL Format*, which is available at:

http://labs.apache.org/webarch/uri/rfc/rfc1959.txt

nms-ldap.properties Configuration File Reference

The `nms-ldap.properties` file contains the settings for communicating with and building LDAP queries to the directory service. This file is located as follows:

- *Windows*: `%NnmInstallDir%\nonOV\jboss\nms\server\nms\conf\props\nms-ldap.properties`
- *UNIX*: `$NnmInstallDir/nonOV/jboss/nms/server/nms/conf/props/nms-ldap.properties`

In the `nms-ldap.properties` file, the following conventions apply:

- To comment out a line, begin that line with a number sign character (`#`).
- To specify a backslash character (`\`), escape the backslash character with an additional backslash character (`\\`).

Table 7 describes the parameters in the `nms-ldap.properties` file.

The initial `nms-ldap.properties` file might not include all of the parameters that are listed in Table 7. Add the parameters that you need.

Table 7 Parameters in the nms-ldap.properties File

Parameter	Description
java.naming.provider.url	The URL for accessing the directory service. The format is the protocol (ldap), followed by the fully-qualified host name of the directory server, optionally followed by the port number. For example: `java.naming.provider.url=ldap://ldap.example.com:389/` If the port number is omitted the following defaults apply: • For non-SSL connections, the default port is `389`. • For SSL connections, the default port is `636`. Configuring this parameter enables LDAP communication between NNMi and the directory service. To disable LDAP communication, comment out this parameter, and then save the file. The configuration in the `nms-ldap.properties` file will be ignored.
java.naming.security.protocol	The connection protocol specification. • If the directory service is configured to use LDAP over SSL, set this parameter to `ssl`. For example: `java.naming.security.protocol=ssl` • If the directory service does not require SSL, leave this parameter commented out. For more information, see Configuring an SSL Connection to the Directory Service on page 92.

Table 7 Parameters in the nms-ldap.properties File (cont'd)

Parameter	Description
bindDN	For a directory service (such as Active Directory) that does not allow anonymous access, identifies the user name for accessing the directory service. Because the password for this user name is stored in plain text in the `nms-ldap.properties` file, select a user name with read-only access to the directory service. For example: `bindDN=region1\\john.doe@example.com`
bindCredential	When `bindDN` is set, specifies the password for the user name that `bindDN` identifies. For example: `bindCredential=PasswordForJohnDoe`
baseCtxDN	Identifies the portion of the directory service domain that stores user records. The format is a comma-separated list of directory service attribute names and values. For example: • `baseCtxDN=CN=Users,DC=ldapserver,DC=example,DC=com` • `baseCtxDN=ou=People,o=example.com` For more information, see User Identification on page 99.
baseFilter	Identifies the user who is signing in to NNMi. The format is the name of the directory service user name attribute and a string that relates the entered user sign-in name to the format of names in the directory service. The user name string contains the expression `{0}` (to denote the user name entered for signin) and any other characters that are needed to match the directory service formatting of user names. • If the user name entered for NNMi signin is the same as the user name stored in the directory service, the value is the replacement expression. For example: – `baseFilter=CN={0}` – `baseFilter=uid={0}` • If the user name entered for NNMi signin is as subset of the user name stored in the directory service, include the additional characters in the value. For example: – `baseFilter=CN={0}@example.com` – `baseFilter=uid={0}@example.com` For more information, see User Identification on page 99.
defaultRole	Optional. Specifies a default role that applies to any directory service user who signs in to NNMi through LDAP. The value of this parameter applies regardless of where role mappings are stored (in the NNMi database or in the directory service). If a user is configured for an NNMi role, NNMi uses the role with more privileges. Valid values are as follows: `admin`, `level2`, `level1`, or `guest`. For example: `defaultRole=guest` If commented out or omitted, no default role is used.

Table 7 Parameters in the nms-ldap.properties File (cont'd)

Parameter	Description
rolesCtxDN	Identifies the portion of the directory service domain that stores group records. The format is a comma-separated list of directory service attribute names and values. For example: • `rolesCtxDN=CN=Users,DC=ldapserver,DC=example,DC=com` • `rolesCtxDN=ou=Groups,o=example.com` In other directory services (not Active Directory), for a faster search, you can identify one or more directory service groups that contain NNMi roles. If the group names form a pattern, you can specify a wildcard. For example, if the directory service includes groups named `USERS-NNMi-administrators`, `USERS-NNMi-level1Operators`, and so forth, you could use a search context similar to: `rolesCtxDN=cn=USERS-NNMi-*,ou=Groups,o=example.com` Configuring this parameter enables directory service queries for NNMi role assignments through LDAP. To disable directory service queries for NNMi role assignments through LDAP, comment out this parameter, and then save the file. The remaining role-related values in the `nms-ldap.properties` file will be ignored. For more information, see Role Identification on page 101.

roleFilter	Identifies the user who is signing in to NNMi, formatted as the directory service stores group member names.
	The format is the name of the directory service group attribute for user ID and a string that relates the entered user sign-in name to the format of user IDs in the directory service. The user name string contains one of the following expressions and any other characters that are needed to match the directory service formatting of group member names.
	• The expression `{0}` denotes the user name entered for signin (for example, `john.doe`).
	An example role filter that matches on the (short) user name entered for signin is:
	`roleFilter=member={0}`
	• The expression `{1}` denotes the distinguished name of the authenticated user as returned by the directory service (for example, `CN=john.doe@example.com,OU=Users,OU=Accounts, DC=example,DC=com` or `uid=john.doe@example.com,ou=People,o=example.com`).
	An example role filter that matches on the (full) authenticated user name is:
	`roleFilter=member={1}`
	For more information, see Role Identification on page 101.

uidAttributeID	Identifies the group attribute that stores the directory service user ID. For example: `uidAttributeID=member` For more information, see Role Identification on page 101.

Table 7 Parameters in the nms-ldap.properties File (cont'd)

Parameter	Description

roleAttributeID	• For Active Directory, identifies the name of the user attribute that stores the user's group membership. For example:
	`roleAttributeID=memberOf`
	• For other directory services, identifies the name of the group attribute that stores the NNMi role that applies to the members of the directory server group. For example:
	`roleAttributeID=nnmiRole`
	For more information, see Role Identification on page 101.

roleAttributeIsDN	Indicates how the user's role attribute is formatted in the directory service.
	• For Active Directory, use:
	`roleAttributeIsDN=true`
	The group's role attribute (specified by `roleNameAttributeID`) represents the distinguished name of a role object, and the role name is taken from the value of the `roleAttributeID` attribute of the corresponding user object.
	• For other directory services, use:
	`roleAttributeIsDN=false`
	The role name is taken directly from the value of the group's role attribute (`roleAttributeID`).
	For more information, see Role Identification on page 101.

roleNameAttributeID	(Active Directory) When `roleAttributeIsDN=true`, identifies the group attribute in the directory service that stores the NNMi role. An example usage is: `roleNameAttributeID=info`
userRoleFilterList	Optional. Limits the NNMi roles whose associated users can be assigned incidents in the NNMi console. The roles in this list apply only to directory service user names authenticated through LDAP. This parameter provides functionality that is not available when NNMi roles are assigned in the NNMi console and stored in the NNMi database. The format is a semicolon-separated list of one or more NNMi role names (as defined in Table 6 on page 101). For example: `userRoleFilterList=admin;level2;level1`
searchTimeLimit	Optional. Specifies the timeout value in milliseconds. The default value is 10000 (10 seconds). If you are encountering timeouts during NNMi user sign in, increase this value. For example: `searchTimeLimit=10000`

Examples

Example nms-ldap.properties file for Active Directory

An example `nms-ldap.properties` file follows for Active Directory:

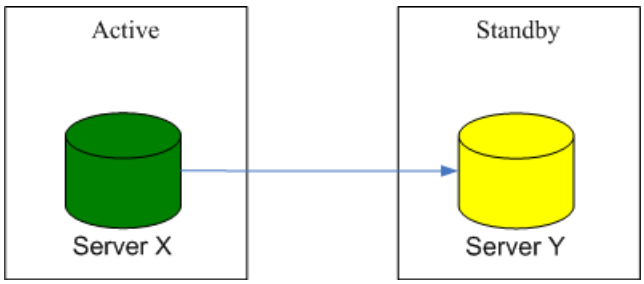
```
java.naming.provider.url=ldap://MYldapserver.example.com:389/
bindDN=MYdomain\\MYusername
bindCredential=MYpassword
baseCtxDN=CN=Users,DC=MYldapserver,DC=EXAMPLE,DC=com
baseFilter=CN={0}
defaultRole=guest
rolesCtxDN=CN=Users,DC=MYldapserver,DC=EXAMPLE,DC=com
roleFilter=member={1}
uidAttributeID=member
roleAttributeIsDN=true
roleNameAttributeID=info
roleAttributeID=memberOf
userRoleFilterList=admin;level2;level1
```

Example nms-ldap.properties file for other directory services

An example `nms-ldap.properties` file follows for other directory services:

```
java.naming.provider.url=ldap://MYldapserver.example.com:389/
baseCtxDN=ou=People,o=EXAMPLE.com
baseFilter=uid={0}
defaultRole=guest
rolesCtxDN=ou=Groups,o=EXAMPLE.com
roleFilter=member={1}
uidAttributeID=member
roleAttributeID=MYnnmiRole
userRoleFilterList=admin;level2;level1
```

Configuring NNMi for Application Failover



Many information technology professionals depend on HP Network Node Manager i Software to notify them when critical network equipment fails and to provide them with a root cause for the failure. They also need NNMi to continue to notify them of network equipment failures, even when the NNMi management server fails. **NNMi application failover** meets this need, transferring application control of NNMi processes from an active NNMi management server to a standby NNMi management server, allowing for continuance of NNMi functionality.

This chapter contains the following topics:

Application Failover Overview

The application failover feature is available for NNMi installations that use the embedded database. After configuring your systems to use the application failover feature, NNMi detects an NNMi management server failure and triggers a secondary server to assume NNMi functionality.

The following terms and definitions apply to configuring NNMi for application failover:

- **jboss Application Server**: An application server program for use with Java 2 Platform, Enterprise Edition (J2EE), and Enterprise Java Beans (EJB).

- **JGroups:** An open source technology for multicast communication and clustering; used internally by jboss for its clustering capabilities.
- **Active**: The server running the NNMi processes.
- **Standby**: The system in the NNMi cluster that is waiting for a failover event; this system is not running NNMi processes.
- **Cluster Member**: A Java process running on a system which is using JGroups technology to connect to a cluster; you can have multiple members on a single system.
- **Postgres**: The embedded database NNMi uses to store information such as topology, incidents, and configuration information.
- **Cluster Manager**: The `nnmcluster` process and tool that is used to monitor and manage the servers for the application failover feature.

Application Failover Basic Setup

To deploy the application failover feature, install NNMi on two servers. This chapter refers to these two NNMi management servers as the **active** and **standby** servers. During normal operation, only the active server is running NNMi services.

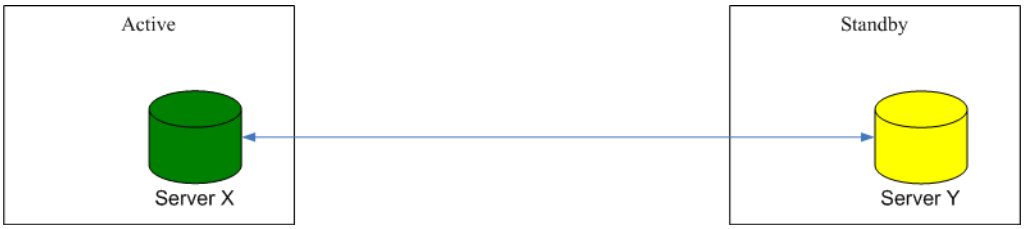
The active and standby NNMi management servers are part of a cluster that monitors a heartbeat signal being generated by both of the NNMi management servers. If the active server fails, resulting in the loss of its heartbeat, then the standby server becomes the active server.

For application failover to work successfully, the following NNMi management server requirements must be met:

- Both NNMi management servers must be running the same type of operating system. For example, if the active server is running an HP-UX operating system, then the standby server must also be running an HP-UX operating system.
- Both NNMi management servers must be running the same NNMi version. For example if NNMi version 8.11 is running on the active server, then the identical NNMi version 8.11 must be on the standby server. The NNMi patch levels must also be the same on both servers.
- The system password must be the same on both NNMi management servers.
- For NNMi installations on Windows operating systems, the `%NnmDataDir%` and `%NnmInstallDir%` system variables must be set to identical values on both servers.
- Both NNMi management servers must have identical licensing attributes. For example, the node counts and licensed features must be identical.
- Do not enable application failover until NNMi is in an advanced stage of initial discovery. For more information see Evaluate Discovery on page 58.
- If the 2 NNMi management servers reside on different subnets, see Application Failover and Multi-Subnets on page 121 to understand how to configure NNMi application failover in a multi-subnet environment.

Configuring NNMi for Application Failover

1 Install NNMi on the active server, server X, and the standby server, server Y, as described in the *NNMi Installation Guide*.



2 Obtain a non-production license for each license on server X and install it on server Y as described in the *License NNMi* section in the *NNMi Installation Guide*.

3 Run the **ovstop** command on each server to shut down NNMi.

4 Configure server X (active) and server Y (standby) for the application failover feature using guidance from the detailed instructions contained in the `ov.conf` file. Use the following procedure:

Edit in the following steps means to uncomment the lines in the text block within the file and to modify the text.

a Edit the following file:

— *Windows*: `%NmmDataDir%\shared\nnm\conf\ov.conf`

— *UNIX*: `$NnmDataDir/shared/nnm/conf/ov.conf`

b Declare a unique name for the NNMi cluster. Use the same name when configuring both the active and standby servers.

```
NNMCLUSTER_NAME=MyCluster
```

c *Optional*. Define other `NNMCLUSTER_*` parameters within the `ov.conf` file. Follow the instructions contained within the `ov.conf` file for modifying each parameter.

5 Select the servers to be the active node (server X) and the standby node (server Y) for the initial run of the cluster. Copy the following file from server X to server Y:

- *Windows*: `%NnmDataDir%\shared\nnm\certificates\nnm.keystore`
- *UNIX*: `$NnmDataDir/shared/nnm/certificates/nnm.keystore`

6 *UNIX*. If server Y is a UNIX server, run the following command:

```
chmod 400 $NnmDataDir/shared/nnm/certificates/nnm.keystore
```

7 Copy the following file from server X to server Y:

- *Windows*: `%NnmDataDir%\shared\nnm\conf\nnmcluster\cluster.keystore`
- *UNIX*: `$NnmDataDir/shared/nnm/conf/nnmcluster/cluster.keystore`

8 *UNIX.* If server Y is a UNIX server, run the following command:

```
chmod 400 \
$NnmDataDir/shared/nnm/conf/nnmcluster/cluster.keystore
```

9 Windows. If servers X and Y are Windows servers, complete the following steps on both servers:

a Open the `Services` menu.

b Locate the `HP Openview Process Manager` service, and change it to `Manual`.

c Locate the `HP NNM Cluster Manage`r service, and change it to `Automatic`.

10 On server X, start the NNMi cluster manager:

```
nnmcluster -daemon
```

This command starts a process that connects the NNMi management server to the cluster, detects that there are no other NNMi management servers present, assumes the active state, starts the NNMi services on the active server, and then creates a database backup.

11 Wait a few minutes to allow server X to become the first active node in the cluster. Run the **`nnmcluster -display`** command on server X and search the displayed results for the term `ACTIVE` as in `ACTIVE_NNM_STARTING` or `ACTIVE_`*SomeOtherState*. Do not continue with step 12 until you know that server X is the active node.

12 On server Y, start the NNMi cluster manager:

```
nnmcluster -daemon
```

13 Instruct NNMi users to store two bookmarks in their browsers, one to server X (the active NNMi server) and one to server Y (the standby NNMi server). If a failover occurs, users can connect to server Y (the standby NNMi server).

14 Instruct network operations center (NOC) personnel to configure their devices to send traps to both server X and server Y. While server X (active) is running, it processes the forwarded traps and server Y (standby) ignores the forwarded traps.

Using the Application Failover Feature

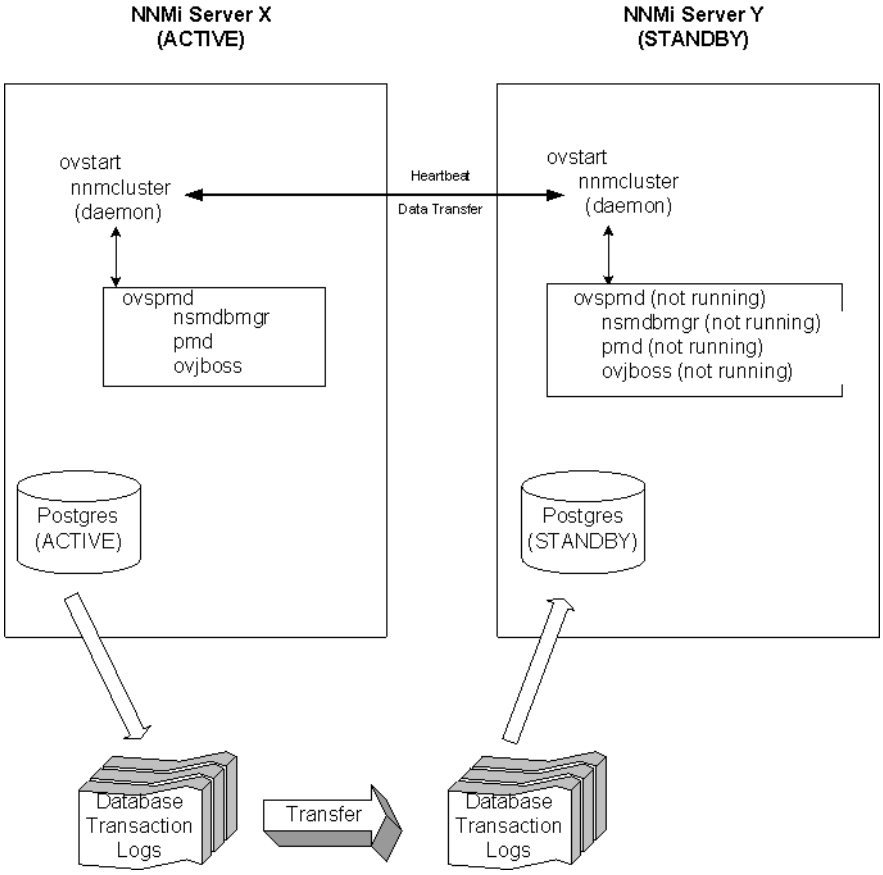
Now that you have both NNMi management servers running the cluster manager, with one active node and one standby node, you can use the cluster manager to view the cluster status. The cluster manager has three modes:

- **daemon mode**: The cluster manager process runs in the background, and uses the `ovstop` and `ovstart` commands to start and stop the NNMi services.
- **interactive mode**: The cluster manager runs an interactive session in which the NNMi administrator can view and change cluster attributes. For example, the NNMi administrator can use this session to enable or disable the application failover feature or shut down the daemon processes.
- **command line mode**: The NNMi administrator views and changes cluster attributes on the command line.

For more information, see the *nmmcluster* reference page, or the UNIX manpage.

Figure 7 shows how the NNMi management servers are configured for the application failover feature. Refer to this figure while reading the rest of this chapter.

Figure 7 Application Failover Configuration



After you start both the active and standby nodes, the standby node detects the active node, requests a database backup from the active node, but does not start NNMi services. This database backup is stored as a single Java-ZIP file. If the standby node already has a ZIP file from a previous cluster-connection, and NNMi finds that the file is already synchronized with the active server, then the file is not retransmitted.

While both the active and standby nodes are running, the active node periodically sends database transaction logs to the standby node. You can modify the frequency of this data transfer by changing the value of the `NNMCLUSTER_DB_ARCHIVE_TIMEOUT` parameter in the `ov.conf` file. These transaction logs accumulate on the standby node, and are available on the standby node any time it needs to become active.

When the standby node receives a full database backup from the primary node, it places the information into its embedded database. It also creates a `recovery.conf` file to inform the embedded database that it should consume all of the received transaction logs before it becomes available to other services.

If the active node becomes unavailable for any reason, the standby node becomes active by running an `ovstart` command, which starts the embedded database, imports the transaction logs, then starts other NNMi services.

If the active NNMi system fails, the standby system synchronizes with its data and transaction log and then begins discovery and polling activities. This transition keeps NNMi monitoring and polling your network while you diagnose and repair the failed system.

Application Failover Scenarios

There are several possible problems that can cause the active NNMi management server to stop sending heartbeats, and to initiate a failover:

- Scenario 1: The active NNMi management server fails.
- Scenario 2: The system administrator shuts down or reboots the active NNMi management server.
- Scenario 3: The NNMi administrator shuts down the cluster.
- Scenario 4: The network connection between the active and the standby NNMi management servers fails.

  In scenario 4, both NNMi management servers run in the active state. When the network device comes back online, the two NNMi management servers auto-negotiate which node should become the new active node.

Additional ovstart and ovstop Options

When you use the `ovstop` and `ovstart` commands on NNMi management servers that have the application failover feature enabled, NNMi actually runs the following commands:

- `ovstart`: **`nnmcluster -daemon`**
- `ovstop`: **`nnmcluster -disable -shutdown`**

The following options to the `ovstop` command apply to NNMi servers configured in an application failover cluster:

- `ovstop -failover`: This command stops the local daemon-mode cluster process and forces a failover to the standby NNMi management server. If the failover mode was previously disabled, then it is re-enabled. This command is equivalent to: `nnmcluster -enable -shutdown`
- `ovstop -nofailover`: This command disables failover mode and then stops the local daemon-mode cluster process. No failover occurs. This command is equivalent to: `nnmcluster -disable -shutdown`
- `ovstop -cluster`: This command stops both the active and standby nodes, removing them both from the cluster. This command is equivalent to: `nnmcluster -halt`

Application Failover Incidents

Any time the `nnmcluster` process or someone using the `nnmcluster` command starts a node as active, NNMi generates one of the following incidents:

- *NnmClusterStartup*: The NNMi cluster was started, and no active node was present. Therefore the node was started in the active state. This incident has a NORMAL severity.

- *NnmClusterFailover*: The NNMi cluster detected a failure of the active node. The standby node was then enabled and NNMi services started on the new active node. This incident has a MAJOR severity.

iSPIs and Application Failover

You can use the application failover feature for a Smart Plug-in (iSPI) that you deploy along with NNMi if the deployment meets the following requirements:

- The iSPI runs on the NNMi management server.
- The iSPI uses the same Postgres instance as NNMi.

The NNM iSPI for Performance is an exception to this description. For information, see Configuring the NNM iSPI for Performance to Work with Application Failover on page 327.

iSPI Installation Information

To install an iSPI on an NNMi management server that is part of an application failover cluster, do the following:

1. As a precaution, run the `nnmconfigexport.ovpl` command on both the active and standby NNMi management servers before proceeding. For information, see Best Practice: Save the Existing Configuration on page 32.
2. As a precaution, back up the NNMi data on both the active and standby NNMi management servers before proceeding. For information, see Backing up NNMi Data on page 183.
3. As a precaution, on the active NNMi management server run the **`nnmcluster -syncdb`** command and wait for the command to complete.
4. On the standby NNMi management server, run the following command:

   `nnmcluster -shutdown` command
5. Edit the following file on the standby NNMi management server.:
   - *Windows*: `%NnmDataDir%\shared\nnm\conf\ov.conf`
   - *UNIX*: `$NnmDataDir/shared/nnm/conf/ov.conf`
6. Comment out the `NNMCLUSTER_NAME` option and save the file.
7. Run the **`ovstart`** command on the standby NNMi management server.
8. Install the iSPI on the standby NNMi management server as described in the iSPI installation guide.
9. Run the **`nnmcluster -halt`** command on the active NNMi management server.
10. Edit the following file on the active NNMi management server:
    - *Windows*: `%NnmDataDir%\shared\nnm\conf\ov.conf`
    - *UNIX*: `$NnmDataDir/shared/nnm/conf/ov.conf`
11. Comment out the `NNMCLUSTER_NAME` option and save the file.

12 Install the iSPI on the active NNMi management server as described in the iSPI installation guide.

13 Edit the following file on both the active and standby NNMi management servers:

- *Windows*: `%NnmDataDir%\shared\nnm\conf\ov.conf`
- *UNIX*: `$NnmDataDir/shared/nnm/conf/ov.conf`

14 Uncommon the `NNMCLUSTER_NAME` option and save each file.

15 Run the **`ovstart`** command on the active NNMi management server.

16 Wait a few minutes to allow the active NNMi management server to become the first active node in the cluster. Run the **`nnmcluster -display`** command on the active NNMi management server and search the displayed results for the term `ACTIVE` as in `ACTIVE_NNM_STARTING` or `ACTIVE_`*SomeOtherState*. Do not continue with step 17 until you know that the active NNMi management server is the active node.

17 Run the **`ovstart`** command on the standby NNMi management server.

Integrated Applications

For products designed to integrate with NNMi, the application failover feature continues to work for NNMi, however the integrating product cannot fail over without some additional configuration.

If the outage appears to be temporary, you can resume using the integrating product after server X is returned to service. To return server X to service, use the following procedure:

1 On server X, run the following command:

```
nnmcluster -daemon
```

Server X joins the cluster and assumes a standby state.

2 On server X, run the following command:

```
nnmcluster -acquire
```

Server X changes to the active state.

If you anticipate that the original server X will be out of service for a longer period of time, you can update the NNMi management server IP address within the integrating product. For instructions on how to modify the IP address field, see the integrating product documentation.

Administrative Tasks and Application Failover

The following information explains how to effectively manage application failover when doing administrative tasks such as patching and restarting NNMi management servers.

Application Failover and NNMi Patches

To apply patches to the NNMi management servers that are configured for application failover, do the following:

1. As a precaution, run the `nnmconfigexport.ovpl` command on both the active and standby NNMi management servers before proceeding. For information, see Best Practice: Save the Existing Configuration on page 32.
2. As a precaution, back up your NNMi data on both the active and standby NNMi management servers before proceeding. For information, see Backing up NNMi Data on page 183.
3. As a precaution, do the following on the active NNMi management server:
   a. Run the **`nnmcluster`** command.
   b. After NNMi prompts you, type **`syncdb`**, then press `Enter`.
   c. Review the displayed information to make sure it includes the following messages:

      ```
      ACTIVE_DB_BACKUP
      ACTIVE_NNM_RUNNING
      STANDBY_READY
      STANDBY_RECV_DBZIP
      STANDBY_READY
      ```
4. Run the **`ovstop`** command on the standby NNMi management server.
5. Run the **`ovstop`** command on the active NNMi management server.
6. Apply the NNMi patch to the active NNMi management server using the instructions provided with the patch.
7. Run the **`ovstart`** command on the active NNMi management server.
8. Verify that the patch installed correctly on the active NNMi management server by viewing information from the **Help** > **About Network Node Manager i-series** menu item located in the NNMi console.
9. Run the **`nnmcluster -syncdb`** command to create a new backup.
10. Apply the NNMi patch to the standby NNMi management server.
11. Run the **`ovstart`** command on the standby NNMi management server.

If you installed the NNM iSPI for Performance, are using the application failover feature, and completed the patch process shown above, run the **`nnmenableperfspi.ovpl`** script on both the active and standby NNMi management servers.

If you are using Linux NNMi management servers, run the following command on both the active and standby NNMi management servers:
`chmod 777 /var/opt/OV/shared/perfSpi/datafiles/nnm_details.xml`

Application Failover and Restarting the NNMi Management Servers

You can restart the standby NNMi management server at any time with no special instructions. If you restart both the standby and active NNMi management servers, restart the primary NNMi management server first.

To restart either the active or the standby NNMi management server, do the following.

1 Run the **nnmcluster -disable** command on the NNMi management server to disable the application failover feature.

2 Restart the NNMi management server.

   a Run the **ovstop** command on the NNMi management server.

   b Run the **ovstart** command on the NNMi management server.

3 Run the **nnmcluster -enable** command on the NNMi management server to enable the application failover feature.

Application Failover and Recovery from a Previous Database Backup

To restore your NNMi database from an original backup when active and standby NNMi servers are configured for application failover, do the following:

1 Run the **ovstop** command on the standby NNMi management server.

2 Run the **ovstop** command on the active NNMi management server.

3 Delete or move the following directory on both the active and standby NNMi management servers:

   - *Windows*: `%NnmDataDir%\shared\nnm\databases\Postgres_standby`
   - *UNIX*: `$NnmDataDir/shared/nnm/databases/Postgres_standby`

4 Restore the database on the primary NNMi management server:

   a Modify the following file to comment out the cluster name:

      — *Windows*: `%NmmDataDir%\shared\nnm\conf\ov.conf`

      — *UNIX*: `$NnmDataDir/shared/nnm/conf/ov.conf`

   b Restore the database as normal. See Restoring NNMi Data on page 186.

   c Run the **ovstop** command on the active NNMi management server.

   d Modify the following file to uncomment the cluster name:

      — *Windows*: `%NmmDataDir%\shared\nnm\conf\ov.conf`

      — *UNIX*: `$NnmDataDir/shared/nnm/conf/ov.conf`

5 Run the **ovstart** command on the active NNMi management server.

6 Wait until the active NNMi management server generates a new backup. To verify that this step is complete, run the **nnmcluster -display** command and look for an `ACTIVE_NNM_RUNNING` message.

7 Run the **ovstart** command on the standby NNMi management server. The standby NNMi management server copies and extracts the new backup. To verify that this step is complete, run the **nnmcluster -display** command and look for a `STANDBY_READY` message.

Application Failover and Multi-Subnets

Overview

HP delivered the NNMi application failover feature with the NNMi 8.11 patch. Future NNMi patches and releases will contain this feature as well.

During the development of the 8.11 release HP did not enable NNMi application failover in a multi-subnet environment. The underlying technology of NNMi application failover uses UDP multicast messages on a single subnet, so it does not route messages across subnets. HP has now enabled NNMi application failover to work in a multi-subnet environment.

This manual section describes the additional setup required to configure NNMi application failover for a multi-subnet environment.

Assumptions

Read the previous sections of this chapter before continuing. Specifically, there are several steps that are required, including:

1. Install NNMi onto two identical servers running the same type of operating system and having the same hardware configuration.
2. Patch both NNMi servers with the same patch-level of the NNMi software.
3. Install identical NNMi licenses (capacity and enabled features) onto these two servers.
4. Copy the `cluster.keystore`, `nnm.keystore` and `nnm.truststore` files from one server to the other.
5. Edit the `ov.conf` file to enable NNMi application failover clustering.

Definitions

This document will use the following terms:

NnmInstallDir – This represents the disk location that contains the NNMi application. You selected this location during NNMi installation. On Unix platforms this location is always `/opt/OV`; on Windows platforms this location is defined at NNMi installation time. Windows servers have a system environment variable, `%NnmInstallDir%`, that points to the correct location.

NnmDataDir – This represents the disk location that contains the NNMi data. On Unix platforms this is `/var/opt/OV`; on Windows platforms this directory is defined at NNMi installation time. Windows servers have a system environment variable, `%NnmDataDir%`, that points to the correct location.

Initial Active – This is the *first* NNMi server that you want to become the active node in the cluster. It is important to identify this machine, as the database on this machine will be replicated to the other NNMi server, the initial standby node.

Detailed Information

As mentioned earlier, the underlying technology of NNMi application failover uses UDP multicast messages for the NNMi servers to discover each other and for the heartbeat signal. NNMi application failover configuration in this multicast environment is very simple: you define a cluster name, manually copy a few files between the nodes, and the nodes automatically discover each other, join the cluster, and exchange database updates.

You can configure NNMi to use TCP, enabling NNMi application failover to work in a multi-subnet environment. This requires only a few additional manual configuration steps:

1. NNMi application failover uses configuration parameters located in the following file: `NnmDataDir/shared/nnm/conf/nnmcluster/jgroupsconfig.xml`

   In the NNMi 8.11 release this file contains the UDP/multicast definitions for NNMi application failover. You must replace this file with the TCP-equivalent configuration file. To do this, complete the following:

   a. On the initial active node, open a shell window or Windows Explorer, and go to the `NnmDataDir/shared/nnm/conf/nnmcluster` directory.

   b. Create a backup copy of the `jgroupsconfig.xml` file in the same directory. For example, you could name this file `jgroupsconfig_udp.xml`.

   c. Copy the `jgroupsconfig.xml` file to `jgroupsconfig_tcp.xml`, or rename the `jgroupsconfig.xml` file to `jgroupsconfig_tcp.xml`.

2. Edit the new TCP-enabled `jgroupsconfig.xml` file:

   a. Locate the `<TCPPING ...` line, located about halfway through the file.

   b. In the `initial_hosts` section, explicitly declare the active and standby servers using either its fully-qualified hostname or IP address. The port is always 7800.

   c. If you intend to install NNM iSPI for Performance onto a third node as described in the NNMi Deployment Guide, declare this hostname or IP address as well.

When completed, the line will look similar to the following:

```
<TCPPING timeout="3000"
initial_hosts="a.myco.com[7800],b.myco.com[7800],c.myco.com[780
0]"
port_range="10"
num_initial_members="3"/>
```

Save the file and close your text editor.

3. Copy your modified `jgroupsconfig.xml` file to the initial standby server, and put it into the `NnmDataDir/shared/nnm/conf/nnmcluster` directory.

At this point the servers have been configured for TCP communication of the heartbeat and other data transfer. The machines do not automatically discover each other since they have been explicitly listed in the configuration file. You are ready to

startup the initial active node first, wait for that to complete the transition to ACTIVE_RUNNING, then startup the initial standby node, as described in earlier sections. Start up the nodes as shown below:

1. On the initial active node, run the **nnmcluster** command to start an interactive NNMi application failover administration session. This session will continuously display updates about the state of the cluster as nodes join or leave the cluster or change status.
2. On the initial standby node, run the **nnmcluster** command to start an interactive NNMi application failover administration session. Review the display to see that both ADMIN processes are in the cluster. This assures you that the two nodes see each other. If both ADMIN processes are not in the cluster, verify the contents of the jgroupsconfig.xml file for the correct settings, including the correct hostname.
3. On the initial active node, run the **nnmcluster -daemon** command. This process immediately returns your prompt, as the daemon or Windows service opens in the background.
4. In the two interactive sessions, you should see the DAEMON nodes join the cluster and the states become ACTIVE_RUNNING after a few minutes.
5. After the initial active node state is ACTIVE_RUNNING, the initial standby node can be started by running the **nnmcluster -daemon** command on that server. This node will connect to the cluster and become the standby node.
6. Run the **quit** command to exit the two interactive administration sessions created in step 1 and step 2 above.

Your servers are now configured for NNMi application failover across subnets.

Network Latency/Bandwidth Considerations

NNMi application failover works by exchanging a continuous heartbeat signal between the nodes in the cluster. It uses this same network channel for exchanging other data files such as the NNMi embedded database, database transaction logs, and other NNMi configuration files. HP recommends using a high performance, low latency connection for NNMi application failover when implementing it over a WAN (wide area network).

NNMi's embedded database can become quite large, and can grow to 1GB or more even though this file is always compressed. Also, NNMi generates hundreds or even thousands of transaction logs during the built-in backup-interval (a configuration parameter which defaults to 24 hours). Each transaction log can be several megabytes up to a maximum size of 16MB (these files are also compressed). Data collected from one of HP's test environments is shown below:

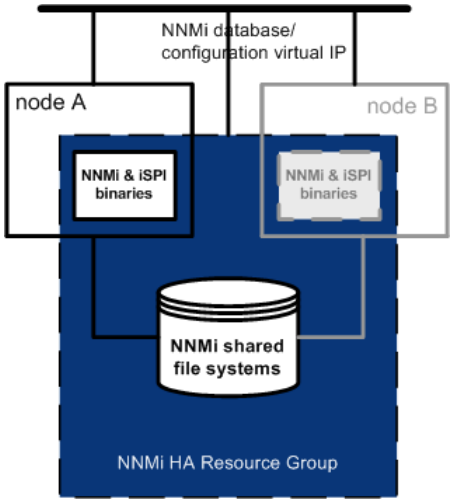
```
Number of nodes managed: 15,000

Number of interfaces: 100,000

Time to complete spiral discovery of all expected nodes: 12 hours

Size of database: 850MB (compressed)

During initial discovery: ~10 transaction logs per minute (peak of
~15/min)

-----------------------------

10 TxLogs/minute X 12 hours = 7200 TxLogs @ ~10MB = ~72GB
```

This is a lot of data to send over the network. If the network between the two nodes is unable to keep up with the bandwidth demands of NNMi application failover then the standby node can fall behind in receiving these database files. This could result in a larger window of potential data loss if the active server fails.

Similarly, if the network between the two nodes has a high latency or poor reliability, this could result in a *false* loss-of-heartbeat between the nodes. For example, this can happen when the heartbeat signal does not respond in a timely manner, and the standby node assumes that the active node has failed. There are several factors involved in detecting loss-of-heartbeat. NNMi avoids false failover notification as long as the network keeps up with the application failover data transfer needs.

In HP's verification of multi-subnet NNMi application failover, the active and standby servers resided in the United States, one in Colorado and another in Virginia. This provided acceptable bandwidth and latency, with no false failovers.

Configuring HP NNM i-series Software in a High Availability Cluster



High availability (HA) refers to a hardware and software configuration that provides for uninterrupted service should some aspect of the running configuration fail. An HA cluster defines a grouping of hardware and software that works together to ensure continuity in functionality and data when failover occurs.

As of HP Network Node Manager i Software version 8.02, NNMi provides support for configuring NNMi to run in an HA cluster under one of several separately-purchased HA products. As of NNMi version 8.10, the NNMi Smart Plug-ins (iSPIs), but not the NNM iSPI NET diagnostics server, can also be running under HA.

This chapter provides a template for configuring NNMi and the iSPIs to run in an HA environment. It does not provide end-to-end instructions for configuring your HA product. The HA configuration commands that NNMi provides are wrappers around the commands for the supported HA products. If you prefer, you can substitute the HA product-specific commands where these instructions specify NNMi-provided commands.

This chapter contains the following topics:

Supported HA Products

The NNMi-provided commands for configuring and running NNMi under HA work with the following HA products for the designated operating systems:

- *HP-UX*: HP ServiceGuard version 11.18 or later
- *Linux*: HP ServiceGuard version 11.18 or later
- *Solaris*: Veritas Cluster Server version 5.0
- *Windows*: Microsoft Cluster Services for Windows 2003

While you can follow the template in this chapter to configure NNMi to run under other HA products, HP does not provide support for cluster configuration issues for other configurations.

Prerequisites to Configuring NNMi for HA

Any system that will be included as a node in an NNMi HA cluster must meet the following requirements:

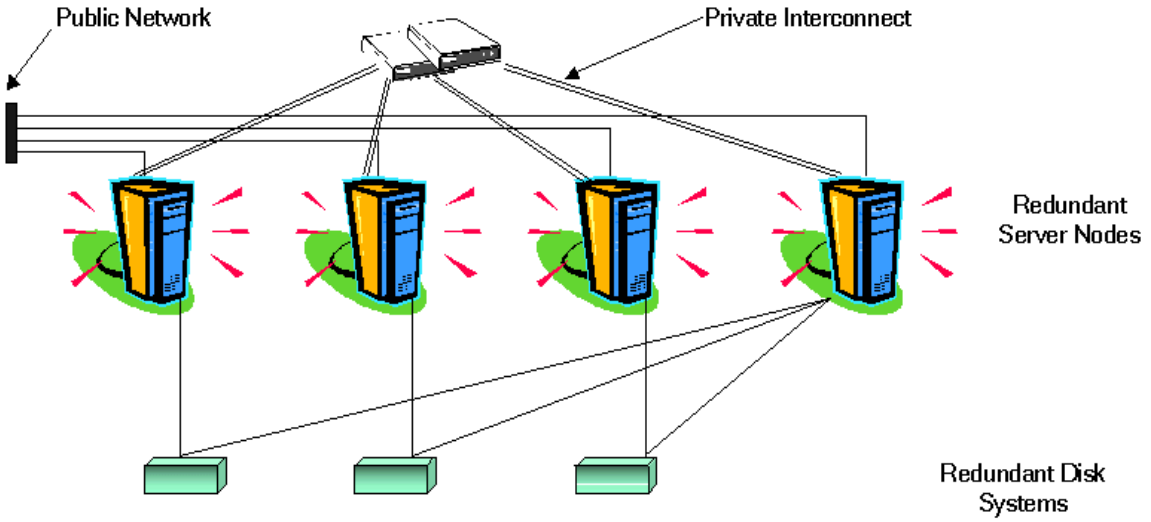
- Supports the use of a virtual IP address.
- Supports the use of a shared disk.
- Meets all of the requirements for NNMi as described in the *HP NNMi Software System and Device Support Matrix*.
- Includes the following additional patches:
  - *Windows*:
    - Microsoft hotfix for *Connecting to SMB share on a Windows 2000-based computer or a Windows Server 2003-based computer may not work with an alias name*, which is available from **http://support.microsoft.com/?id=281308**
  - *UNIX*: No known additional requirements.
- Meets all of the requirements described in the documentation for the HA product on which you plan to run NNMi.

Before you begin to configure NNMi for HA, use the commands for your HA product to configure and test an HA cluster. The HA cluster provides such functionality as heartbeat checking and failover initiation. The HA cluster configuration must, at a minimum, include the following items:

- (UNIX only) ssh
- (UNIX only) remsh
- Virtual IP address for the HA cluster that is DNS-resolvable
- Virtual hostname for the HA cluster that is DNS-resolvable

Cluster architecture provides a single, globally coherent process and resource management view for the multiple nodes of a cluster. Figure 8 shows an example of a cluster architecture.

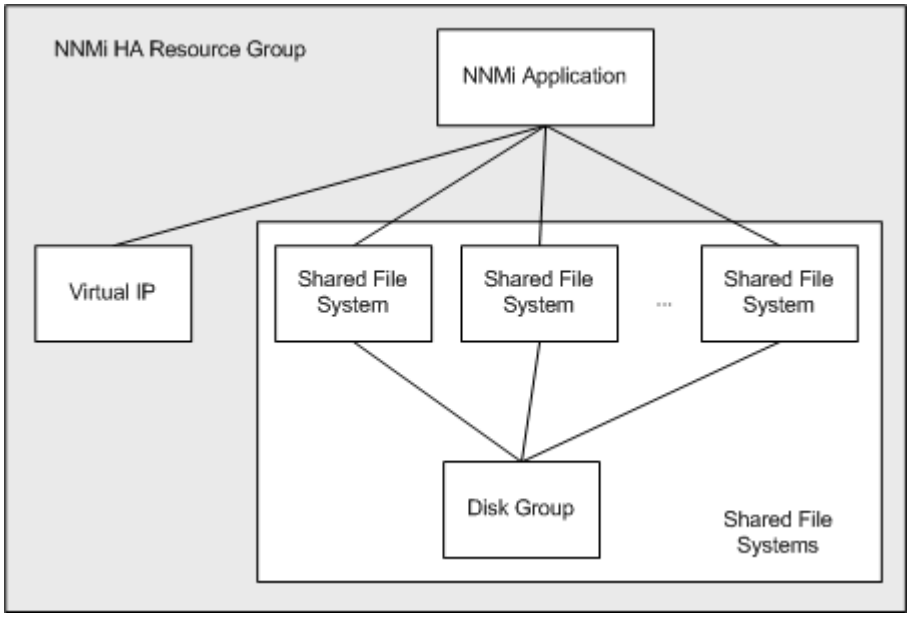
Figure 8 Architecture of a High Availability Cluster



Each node in a cluster is connected to one or more public networks, and to a private interconnect, representing a communication channel for transmitting data between cluster nodes.

In modern cluster environments such as HP ServiceGuard, Veritas Cluster Server, or Microsoft Cluster Services, applications are represented as compounds of resources, which are simple operations that enable applications to run in a cluster environment. The resources construct an **HA resource group**, which represents an application running in a cluster environment. Figure 9 shows an example of how an HA resource group may be organized.

Figure 9 Typical HA Resource Group Layout



This document uses the term *HA resource group* to designate a set of resources in any cluster environment. Each HA product uses a different name for the HA resource group. Table 8 lists the term for each supported HA product that equates to *HA resource group* for this document. (For the specific versions that are supported for each HA product, see Supported HA Products on page 126.)

Table 8 Terminology for HA Resource Group in the Supported HA Products

HA Product	Abbreviation	Equivalent Term for HA Resource Group
HP ServiceGuard	SG	Package
Veritas Cluster Server	VCS	Service Group
Microsoft Cluster Services	MSCS	Resource Group

HA Terms

Table 9 lists and defines some common HA terms.

Table 9 Common HA Terms

Term	Description
HA resource group	An application running in a cluster environment (under an HA product). An HA resource group can simultaneously be a cluster object that represents an application in a cluster.
Volume group	One or more disk drives that are configured to form a single large storage area.
Logical volume	An arbitrary-size space in a volume group that can be used as a separate file system or as a device swap space.
Primary cluster node	The first system on which the software product is installed, *and* the first system on which HA is configured. The shared disk is mounted on the primary cluster node for initial set up. The primary cluster node generally becomes the first active cluster node, but you do not need to maintain the primary designation after HA configuration is complete. The next time you update the HA configuration, another node might become the primary cluster node.
Secondary cluster node	Any system that is added to the HA configuration after the primary cluster node has been fully configured for HA.
Active cluster node	The system that is currently running the HA resource group.
Passive cluster node	Any system that is configured for HA but is not currently running the HA resource group. If the active cluster node fails, the HA resource group fails over to one of the available passive cluster nodes, which then becomes the active cluster node for that HA resource group.

NNMi HA Cluster Scenarios

For NNMi HA configuration, NNMi is installed on each system that will become part of an HA resource group. The NNMi database is located on a separate disk that is accessed by the NNMi programs running on each system. (Only one system, the active cluster node, accesses the shared disk at any given time.)

This approach is valid for the embedded and third-party database solutions.

Run the NNMi database backup and restore scripts on the active cluster node only.

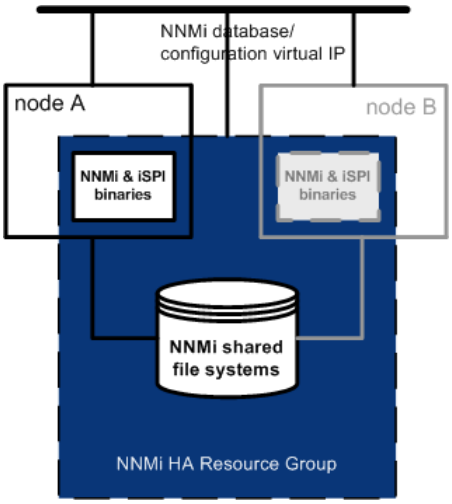
NNMi-only scenario

Figure 10 shows a graphical representation of the NNMi HA cluster scenario. In this figure the NNMi HA resource group is synonymous with the NNMi HA cluster.

Node A and node B are each a fully installed NNMi management server that contains the NNMi program and any iSPIs that run on that system. The active cluster node accesses the shared disk for runtime data. Other products connect to NNMi by means of the virtual IP address of the HA resource group.

If the cluster contains more than two NNMi nodes, additional nodes are configured similarly to node B in Figure 10.

Figure 10 Basic Scenario for NNMi HA Cluster

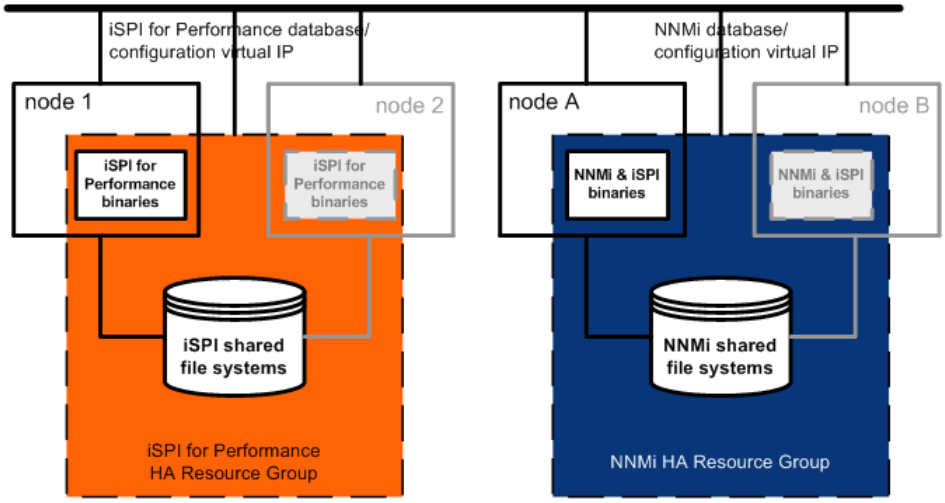


For information about how to implement this scenario, see Configuring NNMi for HA on page 133 and Configuring iSPIs for HA on page 139.

NNMi and NNM iSPI for Performance on a dedicated server scenario

If you are running the NNM iSPI for Performance on a dedicated server, you can configure this iSPI to run as a separate HA resource group within the NNMi HA cluster, as shown in Figure 11. The NNMi HA resource group is as described for the NNMi-only scenario.

Figure 11 HA for NNMi and NNM iSPI for Performance on a Dedicated Server



For information about how to implement this scenario, see Configuring NNMi for HA on page 133, Configuring iSPIs for HA on page 139, and Configuring the NNM iSPI for Performance on a Dedicated Server for HA on page 149.

Other options for the NNM iSPI for Performance on a dedicated server are as follows:

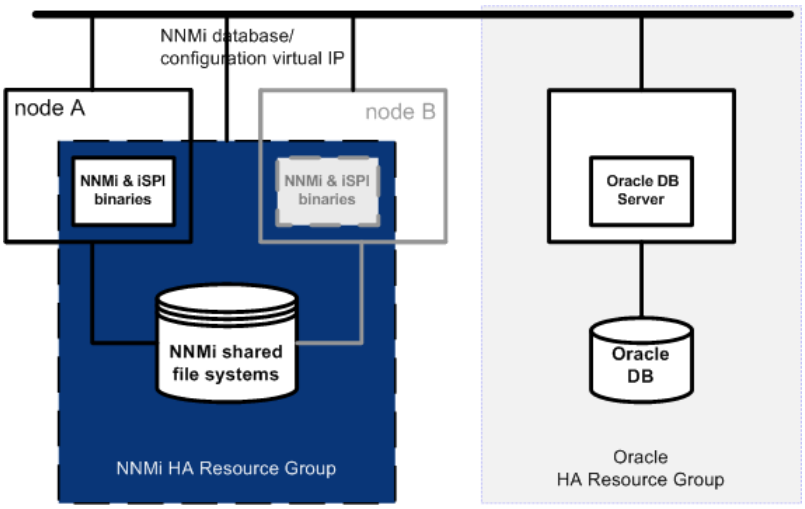
- Run the NNM iSPI for Performance on a single system with no HA. Use this approach while evaluating the iSPI and for environments where it is not critical for performance data to be always available.
- Configure the NNM iSPI for Performance to run under a different HA cluster than that for NNMi. In this case, you must manage the NNM iSPI for Performance's dependency on NNMi manually.

NNMi with an Oracle database scenario

If your NNMi implementation utilizes Oracle for the main NNMi database, the Oracle database should be on a separate server, as shown in Figure 12, for performance reasons. Therefore, you must configure two HA resource groups within the NNMi HA cluster:

- The NNMi HA resource group includes the NNMi nodes and a shared disk for NNMi data that is not stored in the Oracle database.
- The Oracle HA resource group contains the Oracle database server and the database disk.

Figure 12 HA for NNMi with an Oracle Database

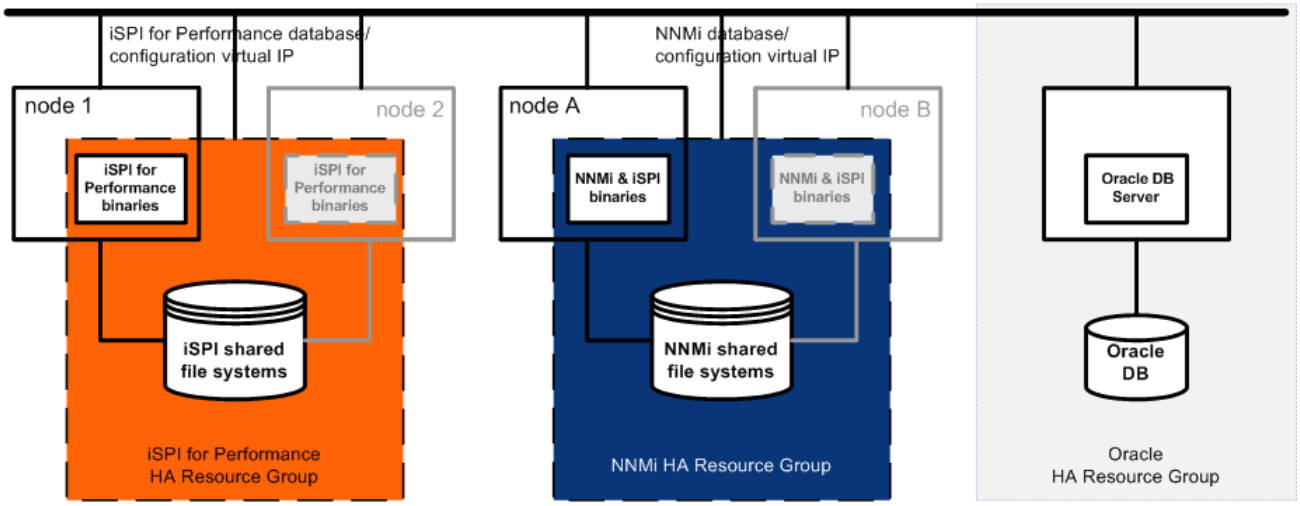


For information about how to implement this scenario, see Configuring NNMi for HA in an Oracle Environment on page 152 and Configuring iSPIs for HA on page 139.

NNMi with an Oracle database and NNM iSPI for Performance on a dedicated server scenario

If your NNMi implementation utilizes Oracle for the main NNMi database and you are running the NNM iSPI for Performance on a dedicated server, you can configure three HA resource groups within the NNMi HA cluster, as shown in Figure 13.

Figure 13 HA for NNMi with an Oracle Database and NNM iSPI for Performance on a Dedicated Server



For information about how to implement this scenario, see Configuring NNMi for HA in an Oracle Environment on page 152, Configuring iSPIs for HA on page 139, and Configuring the NNM iSPI for Performance on a Dedicated Server for HA on page 149.

Manpages

With regard to HA configuration, the NNMi manpages contain the following topics:

- nnm-ha
- nnmhaconfigure.ovpl
- nnmhaunconfigure.ovpl
- nnmhadisk.ovpl
- nnmhaclusterinfo.ovpl
- nnmhastartrg.ovpl
- nnmhastoprg.ovpl

On the Windows operating system, these manpages are available as text files.

Configuring HA

Configuring NNMi for HA

The two distinct phases of configuring NNMi for HA are as follows:

1 Copy the NNMi data files to the shared disk.

- Do this task on the primary node, as described in step 2 through step 12 of Configuring NNMi on the Primary Cluster Node on page 135.

2 Configure NNMi to run under HA.

- Do this task on the primary node, as described in step 13 through step 16 of Configuring NNMi on the Primary Cluster Node on page 135.
- Do this task on the secondary node, as described in Configuring NNMi on the Secondary Cluster Nodes on page 138.

Designate one HA cluster node as the primary NNMi management server. This is the node that you expect to be active most of the time. Configure the primary node; then configure all other nodes in the HA cluster as secondary nodes.

You *cannot* configure NNMi for HA simultaneously on multiple cluster nodes. After the HA configuration process is completed on one cluster node, proceed with the HA configuration on the next node, and so forth until NNMi is configured for HA on all nodes in the cluster environment.

NNMi HA Configuration Information

The HA configuration script collects information about the NNMi HA resource group. Table 10 lists the information that you will need for configuring the primary node. Gather this information before you begin the configuration procedure.

Table 10 NNMi HA Primary Node Configuration Information

HA Configuration Item	Description
HA resource group	The name of the resource group for the HA cluster that contains NNMi. For example: nnmtest1
Virtual host short name	The short name for the virtual host. This hostname must map to the virtual IP address for the HA resource group. The `nslookup` command must be able to resolve the virtual host short name and the virtual IP address. **NOTE:** If NNMi is unable to resolve the virtual host short name or the virtual host IP address, the HA configuration script could leave the system in an unstable state. Therefore, HP recommends that you implement a secondary naming strategy (such as entering the information in the `%SystemRoot%\system32\drivers\etc\hosts` file on the Windows operating system or `/etc/hosts` file on UNIX operating systems) in case DNS is not available during NNMi HA configuration.
Virtual host netmask	The subnet mask that is used with the virtual host IP address.

Table 10 NNMi HA Primary Node Configuration Information

HA Configuration Item	Description
Virtual host network interface	The network interface on which the virtual host IP address is running. For example: • *Windows*: Local Area Connection • *HP-UX*: lan0 • *Linux*: eth0 • *Solaris*: bge0
Shared file system type	The type of shared disk configuration being used for the HA resource group. Possible values are: • `disk`—The shared disk is a physically attached disk that uses a standard file system type. The HA configuration script can configure the shared disk. For more information, see the File system type entry in this table. • `none`—The shared disk uses a configuration other than that described for the `disk` option, such as SAN or NFS. After running the HA configuration script, configure the shared disk as described in Configuring the Shared Disk on page 155.
File system type	The file system type of the shared disk (if the shared file system type is `disk`). The HA configuration scripts pass this value to the HA product so that it can determine how to validate the disk. HP has tested the following shared disk formats: • *Windows*: Basic (see A Note about Shared Disk Configuration on Windows 2003 on page 155) • *HP-UX*: vxfs • *Linux*: lvm2 • *Solaris*: vxfs HP expects that other formats for a physically attached disk work for this HA implementation, and HP provides limited support in analyzing NNMi HA systems with other physically attached disks that use a standard file system type. **NOTE**: If you use a shared disk format that HP has not tested, review the resource group configuration to verify that the values inserted by the NNMi HA configuration scripts are correct.
Disk group	(UNIX only) The name of the disk group for the NNMi shared file system. This name is based on the name of the HA resource group. For example: nnmtest1-dg
Volume group	(UNIX only) The name of the volume group for the NNMi shared file system. This name is based on the name of the HA resource group. For example: nnmtest1-vol
Mount point	The directory location for mounting the NNMi shared disk. This mount point must be consistent between systems. (That is, each node must use the same name for the mount point.) For example: • *Windows*: `S:\` • *UNIX*: `/nnmmount`

Configuring NNMi on the Primary Cluster Node

Complete the following procedure on the primary cluster node.

1. Verify that the system meets all of the requirements specified in Prerequisites to Configuring NNMi for HA on page 126.
2. If you have not already done so, install NNMi (including the latest consolidated patch, if any); then verify that NNMi is working correctly.
3. If you expect to run any iSPIs on this NNMi management server, perform the appropriate installations according to the following information:
   - NNM iSPI for Performance—Do not install this iSPI until after you have completely configured NNMi to run under HA. For more information, see Configuring the NNM iSPI for Performance as an Add-on iSPI on page 140.
   - NNM iSPI Network Engineering Toolset
     - The SNMP trap analytics and troubleshooting tools are automatically installed with NNMi. No extra work is needed to run these tools under HA.
     - The NNM iSPI NET diagnostics server cannot be included in the NNMi HA resource group. Do not install this component on the NNMi management server. If the NNM iSPI NET diagnostics server is already installed on this NNMi management server, uninstall this component before continuing. For information, see About the NNM iSPI Network Engineering Toolset and NNMi Running under HA on page 141.
   - NNM iSPI for MPLS—Install this iSPI now, before configuring NNMi to run under HA.
   - NNM iSPI for IP Multicast—Install this iSPI now, before configuring NNMi to run under HA.
   - NNM iSPI for IP Telephony—Install this iSPI now, before configuring NNMi to run under HA.
4. Back up the NNMi configuration:

   `nnmbackup.ovpl -scope config -target <directory>`

   For more information about this command, see NNMi Backup and Restore Tools on page 181.
5. Back up the NNMi license file by copying the following file to another location:
   - *Windows*: `%AUTOPASS_HOME%\data\LicFile.txt`

     To determine the value of `%AUTOPASS_HOME%`, examine the system environment variables for the computer.
   - *UNIX*: `/var/opt/OV/HPOvLIC/LicFile.txt`

6 Define the disk device group (and logical volume), consisting of at least one shared disk for the NNMi HA resource group. For example:

- *Windows*: Use Disk Management to configure the disk mount point and format the disk.
- *HP-UX* and *Linux*:

```
pvcreate -f /dev/dsk/<physical_disk_name>
mkdir /dev/<disk_group>
mknod c 64 0x00h0 /dev/<disk_group>/group
vgcreate /dev/rdsk/<physical_disk_name> <disk_group_name>
lvcreate -L <disk_size> -n <logical_volume> <disk_group>
newfs /dev/<disk_group>/r<logical_volume>
```

- *Solaris*:

  Use the Symantec Veritas Storage Foundation as described here:

  Use `vxdiskadm` to add and initialize the disk.

  Use `vxassist make` to allocate disks by space.

```
mkfs -F vxfs /dev/vx/dsk/<disk_group>/<logical_volume_group>
```

For UNIX operating systems, a reference web site is:
http://www.unixguide.net/unixguide.shtml

7 Create the directory mount point for the shared disk (for example, `S:\` or `/nnmmount`):

- *Windows*: Use Windows Explorer and Disk Management.
- *UNIX*: Verify that the shared disk directory mount point has been created with `root` as the user, `sys` as the group, and the permissions set to `555`. For example:

```
ls -l /nnmmount
```

8 Mount the shared disk. For example:

- *Windows*: Use Disk Management.
- *HP-UX*:

```
mount /dev/<disk_group>/<logical_volume> /nnmmount
```

- *Linux*:

```
mount /dev/<disk_group>/<logical_volume_group> /nnmmount
```

- *Solaris*:

```
vxfs: mount /dev/vx/dsk/<disk_group>/<volume_group> /nnmmount
```

9 Stop NNMi:

```
ovstop -c
```

10 Copy the NNMi database to the shared disk:

- *Windows*:

  ```
  %NnmInstallDir%\misc\nnm\ha\nnmhadisk.ovpl NNM \
  -to <HA_mount_point>
  ```

- *UNIX*:

  ```
  $NnmInstallDir/misc/nnm/ha/nnmhadisk.ovpl NNM \
  -to <HA_mount_point>
  ```

To prevent database corruption, run this command (with the `-to` option) only one time. For information about alternatives, see Re-Enable NNMi for HA after All Cluster Nodes are Unconfigured on page 174.

11 Unmount the shared disk:

- *Windows*: Use Windows Explorer and Disk Management.
- *UNIX*: **`umount <HA_mount_point>`**

12 (UNIX only) Deactivate the disk group:

```
vgchange -a n <disk_group>
```

13 Verify that NNMi is not running:

```
ovstop -c
```

14 Configure the NNMi HA resource group:

- *Windows*:

  ```
  %NnmInstallDir%\misc\nnm\ha\nnmhaconfigure.ovpl NNM
  ```

- *UNIX*:

  ```
  $NnmInstallDir/misc/nnm/ha/nnmhaconfigure.ovpl NNM
  ```

Table 10 on page 133 describes the information that this command requests.

15 In step 14, what value did you specify for the shared file system type (as described for Shared file system type and File system type in Table 10 on page 133)?

- For type `disk`, the `nnmhaconfigure.ovpl` command configured the shared disk. Continue with step 16.
- For type `none`, configure the shared disk as described in Configuring the Shared Disk on page 155; then continue with step 16.

16 Start the NNMi HA resource group:

- *Windows*:

  ```
  %NnmInstallDir%\misc\nnm\ha\nnmhastartrg.ovpl NNM \
  <resource_group>
  ```

- *UNIX*:

  ```
  $NnmInstallDir/misc/nnm/ha/nnmhastartrg.ovpl NNM \
  <resource_group>
  ```

If NNMi does not start correctly, see Troubleshooting the HA Configuration on page 172.

Now that NNMi is running under HA, *do not* use the `ovstart` and `ovstop` commands for normal operation. Use these commands only when instructed to do so for HA maintenance purposes.

Configuring NNMi on the Secondary Cluster Nodes

Complete the following procedure on one secondary cluster node at a time.

1. If you have not already done so, complete the procedure for Configuring NNMi on the Primary Cluster Node on page 135.
2. Verify that the system meets all of the requirements specified in Prerequisites to Configuring NNMi for HA on page 126.
3. If you have not already done so, install NNMi (including the latest consolidated patch, if any), and then verify that NNMi is working correctly.
4. Install the iSPIs that you installed in step 3 of Configuring NNMi on the Primary Cluster Node on page 135.
5. Create a mount point for the shared disk (for example, `S:\` or `/nnmmount`).

   This mount point must use the same name as the mount point that you created in step 7 of the procedure Configuring NNMi on the Primary Cluster Node.
6. Configure the NNMi HA resource group:
   - *Windows*: **`%NnmInstallDir%\misc\nnm\ha\nnmhaconfigure.ovpl NNM`**
   - *UNIX*: **`$NnmInstallDir/misc/nnm/ha/nnmhaconfigure.ovpl NNM`**

   Supply the HA resource group name when the command requests this information.
7. Verify that the configuration was successful:
   - *Windows*:

     ```
     %NnmInstallDir%\misc\nnm\ha\nnmhaclusterinfo.ovpl \
     -group <resource_group> -nodes
     ```

   - *UNIX*:

     ```
     $NnmInstallDir/misc/nnm/ha/nnmhaclusterinfo.ovpl \
     -group <resource_group> -nodes
     ```

   The command output lists all nodes that have been configured for the specified HA resource group.
8. Optionally, test the configuration by taking the resource group on the primary node offline and then bringing the resource group on the secondary node online.

Configuring iSPIs for HA

This section includes several procedures for configuring an iSPI to run under HA. To determine which procedures are appropriate for your situation, see Table 11. The entries in this table assume that NNMi is already running under HA.

Table 11 Options for Configuring iSPIs to Run under HA

Software	Already Installed on the NNMi Management Server?	Configuration Information
NNM iSPI for Performance	No	• If the NNM iSPI for Performance will run as an add-on to the NNMi management server, see Configuring the NNM iSPI for Performance as an Add-on iSPI on page 140. • If the NNM iSPI for Performance will run on a dedicated server, see Configuring the NNM iSPI for Performance on a Dedicated Server for HA on page 149.
	Yes	Configuring an Installed Add-On iSPI on page 146 Also see the *NNM iSPI for Performance Installation Guide*.
NNM iSPI NET diagnostics server	No	About the NNM iSPI Network Engineering Toolset and NNMi Running under HA on page 141
	Yes	About the NNM iSPI Network Engineering Toolset and NNMi Running under HA on page 141
NNM iSPI for MPLS	No	1 Installing an Add-On iSPI after HA Configuration (All Other iSPIs) on page 142 2 Configuring an Installed Add-On iSPI on page 146
	Yes	Configuring an Installed Add-On iSPI on page 146

Table 11 Options for Configuring iSPIs to Run under HA (cont'd)

Software	Already Installed on the NNMi Management Server?	Configuration Information
NNM iSPI for IP Multicast	No	1 Installing an Add-On iSPI after HA Configuration (All Other iSPIs) on page 142 2 Configuring an Installed Add-On iSPI on page 146
	Yes	Configuring an Installed Add-On iSPI on page 146
NNM iSPI for IP Telephony	No	1 Installing an Add-On iSPI after HA Configuration (All Other iSPIs) on page 142 2 Configuring an Installed Add-On iSPI on page 146
	Yes	Configuring an Installed Add-On iSPI on page 146

Configuring the NNM iSPI for Performance as an Add-on iSPI

NNM iSPI for Performance on the NNMi management server

The simplest procedure for configuring same-system NNM iSPI for Performance to run under the NNMi HA resource group is as follows:

1 Completely configure and start the NNMi HA resource group for all nodes in the NNMi HA cluster as described in Configuring NNMi for HA on page 133 or Configuring NNMi for HA in an Oracle Environment on page 152. Verify that the cluster is properly configured as described in step 7 on page 138.

The tools that configure the NNM iSPI for Performance to run under HA interact with NNMi during the configuration process. For the simplest configuration experience, ensure that all nodes in the NNMi HA cluster are up during the entire iSPI configuration process.

2 On the active node, verify that the NNMi services are running:

```
ovstatus -c
```

All NNMi services should show the state RUNNING.

3 On the active node in the NNMi HA cluster, do the following:

a Run the NNM iSPI for Performance enablement script.

b Install the NNM iSPI for Performance.

For information about how to complete these steps, see the *NNM iSPI for Performance Installation Guide*.

4 On each passive node in the NNMi HA cluster, do the following:

a Run the NNM iSPI for Performance enablement script.

b Install the NNM iSPI for Performance.

About the NNM iSPI Network Engineering Toolset and NNMi Running under HA

The NNM iSPI Network Engineering Toolset SNMP trap analytics and troubleshooting tools are automatically installed with NNMi. No extra work is needed to run these tools under HA.

The NNM iSPI NET diagnostics server cannot be included in the NNMi HA resource group. To run the NNM iSPI NET diagnostics server on a system that is outside of the NNMi HA resource group, follow these steps:

1. Completely configure the NNMi HA resource group.
2. Install the NNM iSPI NET diagnostics server on a system that is outside of the NNMi HA resource group. During the NNM iSPI NET diagnostics server installation process, supply the NNMi resource group virtual hostname as the NNM Server Hostname.

   For more information, see the *NNM iSPI Network Engineering Toolset Planning and Installation Guide*.

If the NNM iSPI NET diagnostics server is already installed on an NNMi management server to be configured to run under HA, uninstall the NNM iSPI NET diagnostics server before configuring NNMi to run under HA.

Uninstalling the NNM iSPI NET diagnostics server removes all existing reports.

It may be possible to save existing reports, as described here, but the following procedure is untested:

1. Use the MySQL Administrator GUI to perform a backup of the existing `nnminet` database.

   The MySQL Administrator GUI is available in the GUI tools download area at **http://dev.mysql.com**.
2. Uninstall the NNM iSPI NET diagnostics server.
3. Configure NNMi to run under HA.
4. Install the NNM iSPI NET diagnostics server on a separate system.
5. Before running any flows, use the MySQL Administrator GUI to recover the `nnminet` database onto the new installation.

Installing an Add-On iSPI after HA Configuration (All Other iSPIs)

The following iSPIs cannot be installed while NNMi is running under HA:

- NNM iSPI for MPLS
- NNM iSPI for IP Multicast
- NNM iSPI for IP Telephony

To install one or more of these iSPIs, temporarily unconfigure HA, install the iSPIs on each node in the HA resource group, and then reconfigure HA as described here.

1. Verify that the NNMi configuration is consistent across all HA nodes:
   - *Windows*:

     ```
     %NnmInstallDir%\misc\nnm\ha\nnmdatareplicator.ovpl NNM
     ```

   - *UNIX*:

     ```
     $NnmInstallDir/misc/nnm/ha/nnmdatareplicator.ovpl NNM
     ```

2. Determine which node in the NNMi HA cluster is active. On any node, run the following command:
   - *Windows*:

     ```
     %NnmInstallDir%\misc\nnm\ha\nnmhaclusterinfo.ovpl \
     -group <resource_group> -state
     ```

   - *UNIX*:

     ```
     $NnmInstallDir/misc/nnm/ha/nnmhaclusterinfo.ovpl \
     -group <resource_group> -state
     ```

3. On each passive node, unconfigure any add-on iSPIs from the HA cluster. For each iSPI, run the following command:
   - *Windows*:

     ```
     %NnmInstallDir%\misc\nnm\ha\nnmhaunconfigure.ovpl NNM \
     -addon <iSPI_PM_Name>
     ```

   - *UNIX*:

     ```
     $NnmInstallDir/misc/nnm/ha/nnmhaunconfigure.ovpl NNM \
     -addon <iSPI_PM_Name>
     ```

   Where *<iSPI_PM_Name>* is the base name of the Perl module that the iSPI installs on the NNMi management server. To determine what value to use for *<iSPI_PM_Name>*, see Table 12 on page 147 or the appropriate iSPI chapter in the Integrations and Plug-ins section.

4. On the active node, unconfigure any add-on iSPIs from the HA cluster. For each iSPI, run the following command:
   - *Windows*:

     ```
     %NnmInstallDir%\misc\nnm\ha\nnmhaunconfigure.ovpl NNM \
     -addon <iSPI_PM_Name>
     ```

   - *UNIX*:

     ```
     $NnmInstallDir/misc/nnm/ha/nnmhaunconfigure.ovpl NNM \
     -addon <iSPI_PM_Name>
     ```

5 On any node in the HA cluster, verify that the add-on iSPIs on all nodes have been unconfigured from the HA cluster:

- *Windows*:

  **%NnmInstallDir%\misc\nnm\ha\nnmhaclusterinfo.ovpl **
  -config NNM -get NNM_ADD_ON_PRODUCTS

- *UNIX*:

  **$NnmInstallDir/misc/nnm/ha/nnmhaclusterinfo.ovpl **
  -config NNM -get NNM_ADD_ON_PRODUCTS

The command output lists the add-on iSPI configurations in the format *<iSPI_PM_Name>*[*hostname_list*]. For example:

```
PerfSPIHA[hostname1, hostname2]
```

If any hostname appears in the output, repeat step 3 and step 4 until this command output indicates that no iSPIs are configured.

6 On each passive node, unconfigure NNMi from the HA cluster:

- *Windows*:

  **%NnmInstallDir%\misc\nnm\ha\nnmhaunconfigure.ovpl NNM **
  <resource_group>

- *UNIX*:

  **$NnmInstallDir/misc/nnm/ha/nnmhaunconfigure.ovpl NNM **
  <resource_group>

This command removes access to the shared disk but does not unconfigure the disk group or the volume group.

7 On each passive node, remove the resource group-specific files:

- *Windows*:

  In Windows Explorer, delete all files in the %NnmDataDir%\hacluster*<resource_group>*\ folder.

- *HP-UX*:

  rm -rf $NnmDataDir/hacluster/*<resource_group>*/*
  rm -rf /etc/cmcluster/*<resource_group>*/*

- *Linux*:

  rm -rf $NnmDataDir/hacluster/*<resource_group>*/*
  rm -rf /usr/local/cmcluster/conf/*<resource_group>*/*

- *Solaris*:

  rm -rf $NnmDataDir/hacluster/*<resource_group>*/*

8 On the active node, disable HA resource group monitoring by creating the following maintenance file:

- *Windows*: %NnmDataDir%\hacluster*<resource_group>*\maintenance
- *UNIX*: $NnmDataDir/hacluster/*<resource_group>*/maintenance

The file can be empty.

9 Stop NNMi:

```
ovstop -c
```

To prevent data corruption, ensure that no instance of NNMi is running and accessing the shared disk.

10 Back up the NNMi database that is on the shared disk:

```
nnmbackup.ovpl -type offline -target <backup_directory>
```

For more information on this command, see NNMi Backup and Restore Tools on page 181.

11 Copy the NNMi files from the shared disk to the node:

- *Windows*:

```
%NnmInstallDir%\misc\nnm\ha\nnmhadisk.ovpl NNM \
-from <HA_mount_point>
```

- *UNIX*:

```
$NnmInstallDir/misc/nnm/ha/nnmhadisk.ovpl NNM \
-from <HA_mount_point>
```

12 Remove all NNMi files and directories from the shared disk:

- *Windows*: Use Windows Explorer to delete all files under the shared disk mount point (`%HA_MOUNT_POINT%`, which is, for example `S:\`).
- *UNIX*:

```
rm -rf $HA_MOUNT_POINT/*
```

13 On the active node, stop the NNMi HA resource group:

- *Windows*:

```
%NnmInstallDir%\misc\nnm\ha\nnmhastoprg.ovpl NNM \
<resource_group>
```

- *UNIX*:

```
$NnmInstallDir/misc/nnm/ha/nnmhastoprg.ovpl NNM \
<resource_group>
```

This command does not remove access to the shared disk. Nor does it unconfigure the disk group or the volume group.

14 On the active node, unconfigure NNMi from the HA cluster:

- *Windows*:

```
%NnmInstallDir%\misc\nnm\ha\nnmhaunconfigure.ovpl NNM \
<resource_group>
```

- *UNIX*:

```
$NnmInstallDir/misc/nnm/ha/nnmhaunconfigure.ovpl NNM \
<resource_group>
```

This command removes access to the shared disk but does not unconfigure the disk group or the volume group.

15 On the active node, remove the resource group-specific files:

- *Windows*:

  In Windows Explorer, delete all files in the `%NnmDataDir%\hacluster\<resource_group>\` folder.

- *HP-UX*:

```
rm -rf $NnmDataDir/hacluster/<resource_group>/*
rm -rf /etc/cmcluster/<resource_group>/*
```

- *Linux*:

```
rm -rf $NnmDataDir/hacluster/<resource_group>/*
rm -rf /usr/local/cmcluster/conf/<resource_group>/*
```

- *Solaris*:

```
rm -rf $NnmDataDir/hacluster/<resource_group>/*
```

16 Delete the maintenance file:

- *Windows*: `%NnmDataDir%\hacluster\<resource_group>\maintenance`
- *UNIX*: `$NnmDataDir/hacluster/<resource_group>/maintenance`

17 Use the appropriate operating system commands to unmount the shared disk. For example:

- *Windows*: Use Windows Explorer.
- *UNIX*: `umount /nnmmount`

18 On the node that was active before unconfiguring NNMi from HA, start NNMi:

```
ovstart -c
```

19 On the node that was active before unconfiguring NNMi from HA, verify that NNMi started correctly:

```
ovstatus -c
```

All NNMi services should show the state RUNNING.

20 On the node that was active before unconfiguring NNMi from HA, install all of the following add-on iSPIs that you expect to run on this NNMi management server.

- NNM iSPI for MPLS
- NNM iSPI for IP Multicast
- NNM iSPI for IP Telephony

21 On the node that was active before unconfiguring NNMi from HA, follow the steps in Configuring NNMi on the Primary Cluster Node on page 135.

You do not need to do the following:

- Define a disk device group and logical volume.
- Create a mount point for the shared disk.
- Configure the shared disk.

22 On each node that was passive before unconfiguring NNMi from HA, install the iSPIs that you installed in step 20.

23 On each node that was passive before unconfiguring NNMi from HA, follow the steps in Configuring NNMi on the Secondary Cluster Nodes on page 138.

You do not need to create a mount point for the shared disk.

The newly installed iSPIs are not yet running under HA. Configure HA as described in Configuring an Installed Add-On iSPI on page 146.

Configuring an Installed Add-On iSPI

The information in this section applies to any iSPI that meets the following requirements:

- The iSPI runs on the NNMi management server.

  If you are running the NNM iSPI for Performance on a separate system, see Configuring the NNM iSPI for Performance on a Dedicated Server for HA on page 149.

- The iSPI uses the same Postgres instance as NNMi (except for the NNM iSPI for Performance, which does not use Postgres).
- The iSPI was installed on the NNMi management server prior to HA configuration.

Caveats

- The NNM iSPI for Performance installation process automatically configures the add-on iSPI to run under the NNMi HA resource group. If you have not yet installed the NNM iSPI for Performance, see Configuring the NNM iSPI for Performance as an Add-on iSPI.
- The following iSPIs cannot be installed while NNMi is running under HA:
  - NNM iSPI for MPLS
  - NNM iSPI for IP Multicast
  - NNM iSPI for IP Telephony

  To install one or more of these iSPIs, temporarily unconfigure HA, install the iSPIs on each node in the HA resource group, and then reconfigure HA. For detailed information, see Installing an Add-On iSPI after HA Configuration (All Other iSPIs) on page 142.

General Information

This configuration process requires the base name of the Perl module that the iSPI installs on the NNMi management server. Table 12 lists the names that were known when this chapter was last updated. To determine what value to use for other iSPIs, see the appropriate iSPI chapter in the Integrations and Plug-ins section.

Table 12 iSPI Perl Module Names

iSPI	Perl Module Base Name
NNM iSPI for Performance	PerfSPIHA
NNM iSPI for MPLS	MPLS
NNM iSPI for IP Multicast	MULTICAST
NNM iSPI for IP Telephony	IPT

iSPI configuration for HA is order independent. If you plan to run multiple iSPIs on the NNMi management server, you can use either of the following process flows:

- Complete both step 3 and step 4 of the following procedure for the first iSPI, and then complete both step 3 and step 4 for the second iSPI, and so forth.
- Complete step 3 of the following procedure for all iSPIs, and then complete step 4 for all iSPIs on one passive cluster node at a time.

Procedure

Previously installed add-on iSPIs

To configure an add-on iSPI to run under the HA resource group, follow these steps:

1. Completely configure and start the NNMi HA resource group for all nodes in the NNMi HA cluster as described in Configuring NNMi for HA on page 133 or Configuring NNMi for HA in an Oracle Environment on page 152. Verify that the cluster is properly configured as described in step 7 on page 138.

   The tools that configure iSPIs to run under HA interact with NNMi during the configuration process. For the simplest configuration experience, ensure that all nodes in the NNMi HA cluster are up during the entire iSPI configuration process.

2. On the active node in the NNMi HA cluster, verify that the NNMi services are running:

   ```
   ovstatus -c
   ```

   All NNMi services should show the state RUNNING.

3. On the active node in the NNMi HA cluster, add the iSPI to the NNMi HA resource group:

   - *Windows*:

     ```
     %NnmInstallDir%\misc\nnm\ha\nnmhaconfigure.ovpl NNM \
     -addon <iSPI_PM_Name>
     ```

   - *UNIX*:

     ```
     $NnmInstallDir/misc/nnm/ha/nnmhaconfigure.ovpl NNM \
     -addon <iSPI_PM_Name>
     ```

   Where *<iSPI_PM_Name>* is the base name of the Perl module that the iSPI installs on the NNMi management server. To determine what value to use for *<iSPI_PM_Name>*, see Table 12 on page 147 or the appropriate iSPI chapter in the Integrations and Plug-ins section.

4 Configure the iSPI for HA on each passive node in the NNMi HA cluster, add the iSPI to the NNMi HA resource group:

- *Windows*:

  **%NnmInstallDir%\misc\nnm\ha\nnmhaconfigure.ovpl NNM **
  -addon *<iSPI_PM_Name>*

- *UNIX*:

  **$NnmInstallDir/misc/nnm/ha/nnmhaconfigure.ovpl NNM **
  -addon *<iSPI_PM_Name>*

5 Verify the configuration:

- *Windows*:

  **%NnmInstallDir%\misc\nnm\ha\nnmhaclusterinfo.ovpl **
  -config NNM -get NNM_ADD_ON_PRODUCTS

- *UNIX*:

  **$NnmInstallDir/misc/nnm/ha/nnmhaclusterinfo.ovpl **
  -config NNM -get NNM_ADD_ON_PRODUCTS

  The command output lists the add-on iSPI configurations in the format *<iSPI_PM_Name>*[*hostname_list*]. For example:

  ```
  PerfSPIHA[hostname1, hostname2]
  ```

Configuring the NNM iSPI for Performance on a Dedicated Server for HA

NNM iSPI for Performance HA Configuration Information

The HA configuration script collects information about the NNM iSPI for Performance HA resource group. Table 13 lists the information that you will need for configuring the primary node. Gather this information before you begin the configuration procedure.

Table 13 NNM iSPI for Performance HA Primary Node Configuration Information

HA Configuration Item	Description
HA resource group	The name of the resource group for the HA cluster that contains the NNM iSPI for Performance. For example: perftest1
Virtual host short name	The short name for the virtual host. This hostname must map to the virtual IP address for the HA resource group. The `nslookup` command must be able to resolve the virtual host short name and the virtual IP address. **NOTE:** If the NNMi is unable to resolve the virtual host short name or the virtual host IP address, the HA configuration script could leave the system in an unstable state. Therefore, HP recommends that you implement a secondary naming strategy (such as entering the information in the `%SystemRoot%\system32\drivers\etc\hosts` file on the Windows operating system or `/etc/hosts` file on UNIX operating systems) in case DNS is not available during NNM iSPI for Performance HA configuration.
Virtual host netmask	The subnet mask that is used with the virtual host IP address.
Virtual host network interface	The network interface on which the virtual host IP address is running. For example: • *Windows*: Local Area Connection • *HP-UX*: lan0 • *Linux*: eth0 • *Solaris*: bge0
Shared file system type	The type of shared disk configuration being used for the HA resource group. Possible values are: • `disk`—The shared disk is a physically attached disk that uses a standard file system type. The HA configuration script can configure the shared disk. For more information, see the File system type entry in this table. • `none`—The shared disk uses a configuration other than that described for the `disk` option, such as SAN or NFS. After running the HA configuration script, configure the shared disk as described in Configuring the Shared Disk on page 155.

Table 13 NNM iSPI for Performance HA Primary Node Configuration Information

HA Configuration Item	Description
File system type	The file system type of the shared disk (if the shared file system type is `disk`). The HA configuration scripts pass this value to the HA product so that it can determine how to validate the disk. HP has tested the following shared disk formats: • *Windows*: Basic (see A Note about Shared Disk Configuration on Windows 2003 on page 155) • *HP-UX*: vxfs • *Linux*: lvm2 • *Solaris*: vxfs HP expects that other formats for a physically attached disk work for this HA implementation, and HP provides limited support in analyzing NNMi HA systems with other physically attached disks that use a standard file system type. **NOTE**: If you use a shared disk format that HP has not tested, review the resource group configuration to verify that the values inserted by the NNMi HA configuration scripts are correct.
Disk group	(UNIX only) The name of the disk group for the NNM iSPI for Performance shared file system. This name is based on the name of the HA resource group. For example: perftest1-dg
Volume group	(UNIX only) The name of the volume group for the NNM iSPI for Performance shared file system. This name is based on the name of the HA resource group. For example: perftest1-vol
Mount point	The directory location for mounting the NNM iSPI for Performance shared disk. This mount point must be consistent between systems. (That is, each node must use the same name for the mount point.) For example: • *Windows*: `P:\` • *UNIX*: `/perfmount`

Configuring the NNM iSPI for Performance on the Primary Cluster Node

Best practice

If you have not already done so, run the NNM iSPI for Performance enablement script on each node in the NNMi HA cluster.

Complete the following procedure on the primary cluster node:

1. If you have not already done so, install the NNM iSPI for Performance. In the **NNM iSPI for Performance Configuration** window, enter the required details, and *do not* start the service.

   For information about the fields in this window, see the *NNM iSPI for Performance Installation Guide*.

2. Define the disk device group (and logical volume), consisting of at least one shared disk for the performance HA resource group. For examples, see step 6 on page 136.

3. Create the directory mount point for the shared disk (for example, `P:\` or `/perfmount`). For examples, see step 7 on page 136.

4 Mount the shared disk. For examples, see step 8 on page 136.

5 Verify that the NNM iSPI for Performance is not running:

- *Windows*:

  ```
  %NnmInstallDir%\NNMPerformanceSPI\bin\statusSPI.ovpl
  %NnmInstallDir%\NNMPerformanceSPI\bin\statusERS.ovpl
  ```

- *UNIX*:

  ```
  $NnmInstallDir/NNMPerformanceSPI/bin/statusSPI.ovpl
  $NnmInstallDir/NNMPerformanceSPI/bin/statusERS.ovpl
  ```

6 Configure the performance HA resource group:

- *Windows*:

  ```
  %NnmInstallDir%\misc\nnm\ha\nnmhaconfigure.ovpl PerfSPIHA
  ```

- *UNIX*:

  ```
  $NnmInstallDir/misc/nnm/ha/nnmhaconfigure.ovpl PerfSPIHA
  ```

  Table 13 on page 149 describes the information that this command requests.

7 In step 6, what value did you specify for the shared file system type (as described for Shared file system type and File system type in Table 13 on page 149)?

- For type `disk`, the `nnmhaconfigure.ovpl` command configured the shared disk. Continue with step 8.
- For type `none`, configure the shared disk as described in Configuring the Shared Disk on page 155; then continue with step 8.

8 Start the performance HA resource group:

- *Windows*:

  ```
  %NnmInstallDir%\misc\nnm\ha\nnmhastartrg.ovpl PerfSPIHA \
  <resource_group>
  ```

- *UNIX*:

  ```
  $NnmInstallDir/misc/nnm/ha/nnmhastartrg.ovpl PerfSPIHA \
  <resource_group>
  ```

  If the NNM iSPI for Performance does not start correctly, see General HA Troubleshooting on page 172.

Configuring the NNM iSPI for Performance on the Secondary Cluster Nodes

Complete the following procedure on one secondary cluster node at a time:

1 If you have not already done so, complete the procedure for Configuring the NNM iSPI for Performance on the Primary Cluster Node on page 150.

2 Create a mount point for the shared disk (for example, `P:\` or `/perfmount`).

This mount point must use the same name as the mount point that you created in step 2 of the procedure Configuring the NNM iSPI for Performance on the Primary Cluster Node.

3 Configure the NNM iSPI for Performance HA resource group:

- *Windows*:

  ```
  %NnmInstallDir%\misc\nnm\ha\nnmhaconfigure.ovpl PerfSPIHA
  ```

- *UNIX*:

  ```
  $NnmInstallDir/misc/nnm/ha/nnmhaconfigure.ovpl PerfSPIHA
  ```

Supply the performance HA resource group name when the command requests this information.

4 Verify that the configuration was successful:

- *Windows*:

  ```
  %NnmInstallDir%\misc\nnm\ha\nnmhaclusterinfo.ovpl \
  -group <resource_group> -nodes
  ```

- *UNIX*:

  ```
  $NnmInstallDir/misc/nnm/ha/nnmhaclusterinfo.ovpl \
  -group <resource_group> -nodes
  ```

The command output lists all nodes that have been configured for the specified HA resource group.

5 Optionally, test the configuration by taking the resource group on the primary node offline and then bringing the resource group on the secondary node online.

Configuring NNMi for HA in an Oracle Environment

This sections presents a high-level overview of the process for configuring NNMi with an Oracle database to run under HA. The number of possible Oracle configurations is large, and the configuration process can vary according to the Oracle release. For the most accurate information about configuring Oracle to run under HA and creating an NNMi dependency on the Oracle HA resource group, see the HA product documentation for information about how that product supports an Oracle database. You can also go to the Oracle web site (**http://www.oracle.com**) for information on the appropriate Oracle configuration for your HA product.

NNMi with Oracle HA Configuration Information

When Oracle and NNMi both run under HA, each product must be in a separate HA resource group. The Oracle resource group must be fully started before the NNMi resource group starts. If both resource groups are in the same HA cluster, you can modify the cluster configuration to set resource group ordering. If the resource groups are in different HA clusters, ensure that the NNMi resource group dependency on the Oracle resource group is met.

The NNMi resource group must include a shared disk for the NNMi data that is not stored in the Oracle database.

Configuring NNMi with Oracle for HA

1 If you plan to run Oracle under HA, complete that configuration first.

2 On the primary NNMi node, install NNMi (including the latest consolidated patch, if any). During installation, use the virtual IP address or hostname for the Oracle HA resource group (if applicable).

3 On the primary NNMi node, configure NNMi to run under HA as described in Configuring NNMi on the Primary Cluster Node on page 135.

4 Set up the NNMi dependency on the Oracle HA resource group.

For specific instructions, see the HA product documentation.

5 On the secondary NNMi node, install NNMi (including the latest consolidated patch, if any). During installation, use the virtual IP address or hostname for the Oracle HA resource group (if applicable).

6 On the secondary NNMi node, configure NNMi to run under HA described in Configuring NNMi on the Secondary Cluster Nodes on page 138.

7 For each additional secondary NNMi node, repeat step 5 and step 6.

Shared NNMi Data

This implementation of NNMi running under HA requires the use of a separate disk for sharing files between all NNMi nodes in the HA cluster.

NNMi implementations that use Oracle as the primary database also require the use of a separate disk for shared data.

Data on the NNMi Shared Disk

This section lists the NNMi data files that are maintained on the shared disk when NNMi is running under HA.

The locations are mapped to the shared disk location as follows:

- *Windows*:
  - `%NnmInstallDir%` maps to `%HA_MOUNT_POINT%\NNM\installDir`
  - `%NnmDataDir%` maps to `%HA_MOUNT_POINT%\NNM\dataDir`
- *UNIX*:
  - `$NnmInstallDir` maps to `$HA_MOUNT_POINT/NNM/installDir`
  - `$NnmDataDir` maps to `$HA_MOUNT_POINT/NNM/dataDir`

The directories that are moved to the shared disk are as follows:

- *Windows*:
  - `%NnmDataDir%\shared\nnm\databases\Postgres`
    The embedded database; present for all implementations.
  - `%NnmDataDir%\log\nnm`
    The NNMi logging directory.
  - `%NnmDataDir%\shared\nnm\databases\eventdb`
    The pmd events database.
  - `%NnmInstallDir%\nonOV\jboss\nms\server\nms\data`
    The transactional store used by ovjboss.
- *UNIX*:
  - `$NnmDataDir/shared/nnm/databases/Postgres`
    The embedded database; present for all implementations.
  - `$NnmDataDir/log/nnm`
    The NNMi logging directory.
  - `$NnmDataDir/shared/nnm/databases/eventdb`
    The pmd events database.
  - `$NnmInstallDir/nonOV/jboss/nms/server/nms/data`
    The transactional store used by ovjboss.

The `nnmhadisk.ovpl` command copies these files to and from the shared disk. Run this command as the instructions in this chapter indicate. For a summary of the command syntax, see the *nnm-ha* manpage.

Configuration File Replication

The NNMi HA implementation uses file replication to maintain copies of the NNMi configuration files on all NNMi nodes in the HA cluster. By default, the NNMi command `nnmdatareplicator.ovpl` manages file replication. This command copies the NNMi configuration files from the active node to a passive node during the failover process. The `nnmdatareplicator.conf` file specifies the NNMi folders and files that are included in data replication.

For information about the data replication process, see the *nnm-ha* manpage.

Configuring the Shared Disk

If the shared disk is of a format that HP has tested (as listed in Table 10 on page 133), the HA configuration script prepares the shared disk, and you can safely ignore this section.

If the shared disk uses a non-tested configuration, such as SAN or NFS, you must prepare the disk manually. Enter the value `none` for the file system type during HA configuration, and then modify the disk configuration file according to the product documentation. For example:

- *HP-UX*:

  `/etc/cmcluster/<resource_group>/<resource_group>.cntl`

- *Linux*:

  `/usr/local/cmcluster/conf/<resource_group>/<resource_group>.cntl`

- *Solaris*: Add disk entries and links to the HA configuration file by using the `/opt/VRTSvcs/bin/hares` command.
- *Windows*: Use Cluster Administrator (`cluadmin.exe`) or the `cluster.exe` command to add resources to the resource group.

A Note about Shared Disk Configuration on Windows 2003

According to Microsoft Knowledge Base article 237853, dynamic disks are not supported in Windows 2003 with Microsoft Cluster Services. To ensure the correct disk configuration, review the information on the following web sites:

- **http://support.microsoft.com/kb/237853**
- **http://www.petri.co.il/difference_between_basic_and_dynamic_disks_in_windows_xp_2000_2003.htm**

Licensing NNMi in an HA Cluster

For NNMi running under HA, the NNMi license is tied to the virtual IP address of the cluster. Therefore, an HA cluster requires only one NNMi license.

To correctly install the NNMi license for an HA cluster, perform these steps on the active NNMi cluster node:

1. At a command prompt, enter the following command:
   - *Windows:* **%NnmInstallDir%\bin\nnmlicense.ovpl NNM -g**
   - *UNIX*: **$NnmInstallDir/bin/nnmlicense.ovpl NNM -g**
2. In the **License Password** dialog box, click **Request License**.
3. Follow the instructions on the screen to obtain the permanent license password for the virtual IP address of the HA cluster.
4. At a command prompt, enter the following command to update your system and to store your license data files:
   - *Windows*:

     %NnmInstallDir%\bin\nnmlicense.ovpl NNM -f *<license_file>*
   - *UNIX*:

     $NnmInstallDir/bin/nnmlicense.ovpl NNM -f *<license_file>*
5. Update the file `licenses.txt` in the `NNM` directory on the shared disk (for example `S:\NNM\licenses.txt` or `/nnmount/NNM/licenses.txt`) with the new information from the license file on the active node, which is in the following location:
   - *Windows*: `%AUTOPASS_HOME%\data\LicFile.txt`

     To determine the value of `%AUTOPASS_HOME%`, examine the system environment variables for the computer.
   - *UNIX*: `/var/opt/OV/HPOvLIC/LicFile.txt`

   Do one of the following:
   - If this file exists on the shared disk, append the new license keys in `LicFile.txt` on the active node to `licenses.txt` on the shared disk.
   - If this file does not exist on the shared disk, copy `LicFile.txt` from the active node to `licenses.txt` in the `NNM` directory on the shared disk.

Maintaining the HA Configuration

Maintenance Mode

Putting an HA Resource Group into Maintenance Mode

Putting an HA resource group into maintenance mode disables HA resource group monitoring. When an HA resource group is in maintenance mode, stopping and starting the products in that HA resource group do not cause failover.

To put an HA resource group into maintenance mode, on the active cluster node, create the following file:

- *Windows*: `%NnmDataDir%\hacluster\<resource_group>\maintenance`
- *UNIX*: `$NnmDataDir/hacluster/<resource_group>/maintenance`

  The file can be empty.

In NNMi 8.0x, the maintenance file is:

- *Windows*: `%NnmDataDir%\hacluster\maint_NNM`
- *UNIX*: `$NnmDataDir/hacluster/<resource_group>/maint_NNM`

Removing an HA Resource Group from Maintenance Mode

Taking an HA resource group out of maintenance mode re-enables HA resource group monitoring. Stopping the products in that HA resource group causes the HA resource group to fail over to a passive cluster node.

To remove an HA resource group from maintenance mode, delete the maintenance file from the node that was the active cluster node before maintenance was initiated. This file is described in Putting an HA Resource Group into Maintenance Mode.

Maintaining NNMi in an HA Cluster

Starting and Stopping NNMi

While NNMi is running under HA, *do not* use the `ovstart` and `ovstop` commands unless instructed to do so for HA maintenance purposes. For normal operation, use the NNMi-provided HA commands or the appropriate HA product commands for starting and stopping resource groups.

Changing NNMi Hostnames and IP Addresses in a Cluster Environment

A node in a cluster environment can have more than one IP address and hostname. If a node becomes a member of another subnet, you might need to change its IP addresses. As a result, the IP address or fully-qualified domain name might change.

For example, on UNIX systems, the IP address and the related hostname are generally configured in one of the following:

- /etc/hosts
- Domain Name Service (DNS)
- Network Information Service (NIS on HP-UX or Linux, NIS+ on Solaris)

NNMi also configures the hostname and IP address of the management server for the managed node in the NNMi database.

If you are moving from a non-name-server environment to a name-server environment (that is, DNS or BIND), make sure that the name server can resolve the new IP address.

Hostnames work within IP networks to identify a managed node. While a node may have many IP addresses, the hostname is used to pinpoint a specific node. The system hostname is the string returned when you use the `hostname` command.

To change the virtual hostname or IP address of the management server, perform these steps on the active NNMi cluster node:

1. At a command prompt, enter the following command:
   - *Windows:* **`%NnmInstallDir%\bin\nnmlicense.ovpl NNM -g`**
   - *UNIX*: **`$NnmInstallDir/bin/nnmlicense.ovpl NNM -g`**
2. In the **License Password** dialog box, click **Request License**.
3. Follow the instructions on the screen to obtain the permanent license password for the new virtual IP address of the HA cluster.
4. At a command prompt, enter the following command to update your system and to store your license data files:
   - *Windows*:

     `%NnmInstallDir%\bin\nnmlicense.ovpl NNM -f <license_file>`
   - *UNIX*:

     `$NnmInstallDir/bin/nnmlicense.ovpl NNM -f <license_file>`
5. Update the file `licenses.txt` in the `NNM` directory on the shared disk (for example `S:\NNM\licenses.txt` or `/nnmount/NNM/licenses.txt`) with the new information from the license file on the active node, which is in the following location:
   - *Windows*: `%AUTOPASS_HOME%\data\LicFile.txt`

     To determine the value of `%AUTOPASS_HOME%`, examine the system environment variables for the computer.
   - *UNIX*: `/var/opt/OV/HPOvLIC/LicFile.txt`

   Do one of the following:
   - If this file exists on the shared disk, append the new license keys in `LicFile.txt` on the active node to `licenses.txt` on the shared disk.
   - If this file does not exist on the shared disk, copy `LicFile.txt` from the active node to `licenses.txt` in the `NNM` directory on the shared disk.
6. Put the HA resource group into maintenance mode as described in Putting an HA Resource Group into Maintenance Mode on page 157.

7 Stop NNMi:

```
ovstop -c
```

8 Change the IP address or node name of the NNMi management server:

a In the `ov.conf` file, edit the `NNM_INTERFACE` entry to be the new hostname or IP address.

b In the `ovspmd.auth` file, edit any lines containing the old hostname to contain the new hostname.

The `ov.conf` and `ovspmd.auth` files are available in the following location:

- *Windows*: `%NnmDataDir%\shared\nnm\conf`
- *UNIX*: `$NnmDataDir/shared/nnm/conf`

9 Set the cluster configuration:

a Stop the NNMi HA resource group:

— *Windows*:

```
%NnmInstallDir%\misc\nnm\ha\nnmhastoprg.ovpl NNM \
<resource_group>
```

— *UNIX*:

```
$NnmInstallDir/misc/nnm/ha/nnmhastoprg.ovpl NNM \
<resource_group>
```

b Change the cluster configuration to use the new IP address:

— *Microsoft Cluster Services*:

In Cluster Administrator, open ***<resource_group>***.

Double-click ***<resource_group>*-ip**, select **Parameters**, and the enter the new IP address.

— *HP ServiceGuard*: On the active HA cluster node, edit the following file:

HP-UX: `/etc/cmcluster/<resource_group>/<resource_group>.cntl`

Linux: `/usr/local/cmcluster/conf/<resource_group>/<resource_group>.cntl`

Replace `IP[0]=<old_IP_address>` with `IP[0]=<new_IP_address>`. Then use `cmapplyconf` to update all other systems.

— *VERITAS Cluster Server*:

```
$NnmInstallDir/misc/nnm/ha/nnmhargconfigure.ovpl NNM \
<resource_group> -set_value <resource_group>-ip \
Address <new_IP_address>
```

c Start the NNMi HA resource group:

— *Windows*:

```
%NnmInstallDir%\misc\nnm\ha\nnmhastartrg.ovpl NNM \
<resource_group>
```

— *UNIX*:

```
$NnmInstallDir/misc/nnm/ha/nnmhastartrg.ovpl NNM \
<resource_group>
```

10 Verify that NNMi started correctly:

```
ovstatus -c
```

All NNMi services should show the state RUNNING.

11 Take the HA resource group out of maintenance mode as described in Removing an HA Resource Group from Maintenance Mode on page 157.

Stopping NNMi Without Causing Failover

When you need to perform NNMi maintenance, you can stop NNMi on the active cluster node without causing failover to a currently passive node. Follow these steps on the active cluster node:

1 Put the HA resource group into maintenance mode as described in Putting an HA Resource Group into Maintenance Mode on page 157.

2 Stop NNMi:

```
ovstop -c
```

Restarting NNMi after Maintenance

If you have stopped NNMi in the manner that prevents failover, follow these steps to restart NNMi and HA monitoring:

1 Start NNMi:

```
ovstart -c
```

2 Verify that NNMi started correctly:

```
ovstatus -c
```

All NNMi services should show the state RUNNING.

3 Take the HA resource group out of maintenance mode as described in Removing an HA Resource Group from Maintenance Mode on page 157.

Maintaining Add-on iSPIs in an NNMi HA Cluster

The iSPIs are closely linked to NNMi. When add-on iSPIs are installed on the nodes in the NNMi HA cluster, use the NNMi HA cluster maintenance procedures as written.

Maintaining the NNM iSPI for Performance on a Dedicated Server in an HA Cluster

Starting and Stopping the NNM iSPI for Performance

While the NNM iSPI for Performance is running under HA, *do not* use the `startSPI`, `startERS`, `stopSPI`, and `stopERS` commands unless instructed to do so for HA maintenance purposes. Also, do *not* use the **Start** and **Stop** buttons in the NNM iSPI for Performance user interface. For normal operation, use the NNMi-provided HA commands or the appropriate HA product commands for starting and stopping resource groups.

Changing the NNM iSPI for Performance System in a Cluster Environment

Follow the procedure for Changing NNMi Hostnames and IP Addresses in a Cluster Environment on page 157, making the appropriate adjustments for working with the NNM iSPI for Performance.

Stopping NNM iSPI for Performance Without Causing Failover

When you need to perform NNM iSPI for Performance maintenance, you can stop NNM iSPI for Performance on the active node without causing failover to a currently passive node. Follow these steps on the active node:

1. Put the HA resource group into maintenance mode as described in Putting an HA Resource Group into Maintenance Mode on page 157.

   The file can be empty.

2. Stop the NNM iSPI for Performance:

   - *Windows*:

     ```
     %NnmInstallDir%\NNMPerformanceSPI\bin\stopSPI.ovpl
     %NnmInstallDir%\NNMPerformanceSPI\bin\stopERS.ovpl
     ```

   - *UNIX*:

     ```
     $NnmInstallDir/NNMPerformanceSPI/bin/stopSPI.ovpl
     $NnmInstallDir/NNMPerformanceSPI/bin/stopERS.ovpl
     ```

Restarting NNM iSPI for Performance after Maintenance

If you have stopped NNM iSPI for Performance in the manner that prevents failover, follow these steps to restart NNM iSPI for Performance and HA monitoring:

1. Start the NNM iSPI for Performance:

   - *Windows*:

     ```
     %NnmInstallDir%\NNMPerformanceSPI\bin\startSPI.ovpl
     %NnmInstallDir%\NNMPerformanceSPI\bin\startERS.ovpl
     ```

   - *UNIX*:

     ```
     $NnmInstallDir/NNMPerformanceSPI/bin/startSPI.ovpl
     $NnmInstallDir/NNMPerformanceSPI/bin/startERS.ovpl
     ```

2. Verify that NNM iSPI for Performance started correctly:

   - *Windows*:

     ```
     %NnmInstallDir%\NNMPerformanceSPI\bin\statusSPI.ovpl
     %NnmInstallDir%\NNMPerformanceSPI\bin\statusERS.ovpl
     ```

   - *UNIX*:

     ```
     $NnmInstallDir/NNMPerformanceSPI/bin/statusSPI.ovpl
     $NnmInstallDir/NNMPerformanceSPI/bin/statusERS.ovpl
     ```

3. Take the HA resource group out of maintenance mode as described in Removing an HA Resource Group from Maintenance Mode on page 157.

Unconfiguring HA

Unconfiguring NNMi from an HA Cluster

The process of removing an NNMi node from an HA cluster involves undoing the HA configuration for that instance of NNMi. You can then run that instance of NNMi as a standalone management server, or you can uninstall NNMi from that node.

If you want to keep NNMi configured for high availability, the HA cluster must contain one node that is actively running NNMi and at least one passive NNMi node. If you want to completely remove NNMi from the HA cluster, unconfigure the HA functionality on all nodes in the cluster.

To completely unconfigure NNMi from an HA cluster, follow these steps:

1 Determine which node in the HA cluster is active. On any node, run the following command:

- *Windows*:

  **%NnmInstallDir%\misc\nnm\ha\nnmhaclusterinfo.ovpl **
  -group *<resource_group>* -state

- *UNIX*:

  **$NnmInstallDir/misc/nnm/ha/nnmhaclusterinfo.ovpl **
  -group *<resource_group>* -state

2 On each passive node, unconfigure any add-on iSPIs from the HA cluster. For each iSPI, run the following command:

- *Windows*:

  **%NnmInstallDir%\misc\nnm\ha\nnmhaunconfigure.ovpl NNM **
  -addon *<iSPI_PM_Name>*

- *UNIX*:

  **$NnmInstallDir/misc/nnm/ha/nnmhaunconfigure.ovpl NNM **
  -addon *<iSPI_PM_Name>*

  Where *<iSPI_PM_Name>* is the base name of the Perl module that the iSPI installs on the NNMi management server. To determine what value to use for *<iSPI_PM_Name>*, see Table 12 on page 147 or the appropriate iSPI chapter in the Integrations and Plug-ins section.

3 On any node in the HA cluster, verify that the add-on iSPIs on all passive nodes have been unconfigured from the HA cluster:

- *Windows*:

  **%NnmInstallDir%\misc\nnm\ha\nnmhaclusterinfo.ovpl **
  -config NNM -get NNM_ADD_ON_PRODUCTS

- *UNIX*:

  **$NnmInstallDir/misc/nnm/ha/nnmhaclusterinfo.ovpl **
  -config NNM -get NNM_ADD_ON_PRODUCTS

  The command output lists the add-on iSPI configurations in the format *<iSPI_PM_Name>*[*hostname_list*]. For example:

  ```
  PerfSPIHA[hostname1, hostname2]
  ```

At this time, only the active node hostname should appear in the output. If a passive node hostname appears in the output, repeat step 2 until this command output includes only the active node hostname.

4 On each passive node, unconfigure NNMi from the HA cluster:

- *Windows*:

```
%NnmInstallDir%\misc\nnm\ha\nnmhaunconfigure.ovpl NNM \
<resource_group>
```

- *UNIX*:

```
$NnmInstallDir/misc/nnm/ha/nnmhaunconfigure.ovpl NNM \
<resource_group>
```

This command removes access to the shared disk but does not unconfigure the disk group or the volume group.

5 On each passive node, remove the resource group-specific files:

- *Windows*:

  In Windows Explorer, delete all files in the `%NnmDataDir%\hacluster\<resource_group>\` folder.

- *HP-UX*:

```
rm -rf $NnmDataDir/hacluster/<resource_group>/*
rm -rf /etc/cmcluster/<resource_group>/*
```

- *Linux*:

```
rm -rf $NnmDataDir/hacluster/<resource_group>/*
rm -rf /usr/local/cmcluster/conf/<resource_group>/*
```

- *Solaris*:

```
rm -rf $NnmDataDir/hacluster/<resource_group>/*
```

If you want to keep NNMi configured for high availability, stop here.

6 On the active node, unconfigure any add-on iSPIs from the HA cluster. For each iSPI, run the following command:

- *Windows*:

```
%NnmInstallDir%\misc\nnm\ha\nnmhaunconfigure.ovpl NNM \
-addon <iSPI_PM_Name>
```

- *UNIX*:

```
$NnmInstallDir/misc/nnm/ha/nnmhaunconfigure.ovpl NNM \
-addon <iSPI_PM_Name>
```

7 On any node in the HA cluster, verify that the add-on iSPIs on all nodes have been unconfigured from the HA cluster:

- *Windows*:

```
%NnmInstallDir%\misc\nnm\ha\nnmhaclusterinfo.ovpl \
-config NNM -get NNM_ADD_ON_PRODUCTS
```

- *UNIX*:

```
$NnmInstallDir/misc/nnm/ha/nnmhaclusterinfo.ovpl \
-config NNM -get NNM_ADD_ON_PRODUCTS
```

If any hostname appears in the output, repeat step 6 until this command output indicates that no iSPIs are configured.

8 On the active node, stop the NNMi HA resource group:

- *Windows*:

  **%NnmInstallDir%\misc\nnm\ha\nnmhastoprg.ovpl NNM **
  <resource_group>

- *UNIX*:

  **$NnmInstallDir/misc/nnm/ha/nnmhastoprg.ovpl NNM **
  <resource_group>

This command does not remove access to the shared disk. Nor does it unconfigure the disk group or the volume group.

9 On the active node, unconfigure NNMi from the HA cluster:

- *Windows*:

  **%NnmInstallDir%\misc\nnm\ha\nnmhaunconfigure.ovpl NNM **
  <resource_group>

- *UNIX*:

  **$NnmInstallDir/misc/nnm/ha/nnmhaunconfigure.ovpl NNM **
  <resource_group>

This command removes access to the shared disk but does not unconfigure the disk group or the volume group.

10 On the active node, remove the resource group-specific files:

- *Windows*:

  In Windows Explorer, delete all files in the `%NnmDataDir%\hacluster\`*`<resource_group>`*`\` folder.

- *HP-UX*:

  rm -rf $NnmDataDir/hacluster/*<resource_group>*/*
  rm -rf /etc/cmcluster/*<resource_group>*/*

- *Linux*:

  rm -rf $NnmDataDir/hacluster/*<resource_group>*/*
  rm -rf /usr/local/cmcluster/conf/*<resource_group>*/*

- *Solaris*:

  rm -rf $NnmDataDir/hacluster/*<resource_group>*/*

11 Unmount the shared disk.

- If you want to reconfigure the NNMi HA cluster at some point, you can keep the disk in its current state.
- If you want to use the shared disk for another purpose, copy all data that you want to keep (as described in the next procedure), and then use the HA product commands to unconfigure the disk group and volume group.

If you want to run NNMi outside of HA on any node with the existing database, follow these steps:

1 On the active node (if one still exists), verify that NNMi is not running:

```
ovstop
```

Alternatively, check the status of the `ovspmd` process by using Task Manager (Windows) or the `ps` command (UNIX).

2 On the current node (where you want to run NNMi outside of HA), verify that NNMi is not running:

```
ovstop
```

To prevent data corruption, ensure that no instance of NNMi or the NNM iSPI for Performance is running and accessing the shared disk.

3 (UNIX only) Activate the disk group:

```
vgchange -a e <disk_group>
```

4 Use the appropriate operating system commands to mount the shared disk. For example:

- *Windows*: Use Windows Explorer.
- *UNIX*: `mount /dev/vgnnm/lvnnm /nnmmount`

5 Copy the NNMi files from the shared disk to the node:

- *Windows*:

```
%NnmInstallDir%\misc\nnm\ha\nnmhadisk.ovpl NNM \
-from <HA_mount_point>
```

- *UNIX*:

```
$NnmInstallDir/misc/nnm/ha/nnmhadisk.ovpl NNM \
-from <HA_mount_point>
```

6 Use the appropriate operating system commands to unmount the shared disk. For example:

- *Windows*: Use Windows Explorer.
- *UNIX*: `umount /nnmmount`

7 (UNIX only) Deactivate the disk group:

```
vgchange -a n <disk_group>
```

8 Start NNMi:

```
ovstart -c
```

NNMi is now running with a copy of the database that was formerly used by the NNMi HA resource group. Manually remove from the NNMi configuration any nodes that you do not want to manage from this NNMi management server.

Unconfiguring the NNM iSPI for Performance from an HA Cluster

The process of removing an NNM iSPI for Performance node from an HA cluster involves undoing the HA configuration for that instance of the iSPI. You can then run that instance of the iSPI as a standalone server, or you can uninstall the iSPI from that node.

If you want to keep the NNM iSPI for Performance configured for high availability, the HA cluster must contain one node that is actively running the iSPI and at least one passive iSPI node. If you want to completely remove the NNM iSPI for Performance from the HA cluster, unconfigure the HA functionality on all nodes in the cluster.

To completely unconfigure the NNM iSPI for Performance from an HA cluster, follow these steps:

1 Determine which node in the HA cluster is active:

- *Windows*:

```
%NnmInstallDir%\misc\nnm\ha\nnmhaclusterinfo.ovpl \
-group <resource_group> -state
```

- *UNIX*:

```
$NnmInstallDir/misc/nnm/ha/nnmhaclusterinfo.ovpl \
-group <resource_group> -state
```

2 On each passive node, unconfigure the NNM iSPI for Performance from the HA cluster:

- *Windows*:

```
%NnmInstallDir%\misc\nnm\ha\nnmhaunconfigure.ovpl \
PerfSPIHA <resource_group>
```

- *UNIX*:

```
$NnmInstallDir/misc/nnm/ha/nnmhaunconfigure.ovpl \
PerfSPIHA <resource_group>
```

This command removes access to the shared disk but does not unconfigure the disk group or the volume group.

3 On each passive node, remove the resource group-specific files:

- *Windows*:

In Windows Explorer, delete all files in the `%NnmDataDir%\hacluster\<resource_group>\` folder.

- *HP-UX*:

```
rm -rf $NnmDataDir/hacluster/<resource_group>/*
rm -rf /etc/cmcluster/<resource_group>/*
```

- *Linux*:

```
rm -rf $NnmDataDir/hacluster/<resource_group>/*
rm -rf /usr/local/cmcluster/conf/<resource_group>/*
```

- *Solaris*:

```
rm -rf $NnmDataDir/hacluster/<resource_group>/*
```

If you want to keep the NNM iSPI for Performance configured for high availability, stop here.

4 On the active node, stop the performance HA resource group:

- *Windows*:

  %NnmInstallDir%\misc\nnm\ha\nnmhastoprg.ovpl PerfSPIHA \ *<resource_group>*

- *UNIX*:

  $NnmInstallDir/misc/nnm/ha/nnmhastoprg.ovpl PerfSPIHA \ *<resource_group>*

  This command does not remove access to the shared disk. Nor does it unconfigure the disk group or the volume group.

5 On the active node, remove the resource group-specific files:

- *Windows*:

  In Windows Explorer, delete all files in the `%NnmDataDir%\hacluster\<resource_group>\` folder.

- *HP-UX*:

  rm -rf $NnmDataDir/hacluster/*<resource_group>*/*
  rm -rf /etc/cmcluster/*<resource_group>*/*

- *Linux*:

  rm -rf $NnmDataDir/hacluster/*<resource_group>*/*
  rm -rf /usr/local/cmcluster/conf/*<resource_group>*/*

- *Solaris*:

  rm -rf $NnmDataDir/hacluster/*<resource_group>*/*

6 Unmount the shared disk.

- If you want to reconfigure the performance HA cluster at some point, you can keep the disk in its current state.
- If you want to use the shared disk for another purpose, copy all data that you want to keep (as described in the next procedure), and then use the HA product commands to unconfigure the disk group and volume group.

If you want to run the NNM iSPI for Performance outside of HA on any node with the existing database, follow these steps:

1 On the active node (if one still exists), verify that the NNM iSPI for Performance is not running:

- *Windows*:

  %NnmInstallDir%\NNMPerformanceSPI\bin\stopSPI.ovpl
  %NnmInstallDir%\NNMPerformanceSPI\bin\stopERS.ovpl

- *UNIX*:

  $NnmInstallDir/NNMPerformanceSPI/bin/stopSPI.ovpl
  $NnmInstallDir/NNMPerformanceSPI/bin/stopERS.ovpl

2 On the current node (where you want to run the NNM iSPI for Performance outside of HA), verify that the iSPI is not running:

- *Windows*:

```
%NnmInstallDir%\NNMPerformanceSPI\bin\stopSPI.ovpl
%NnmInstallDir%\NNMPerformanceSPI\bin\stopERS.ovpl
```

- *UNIX*:

```
$NnmInstallDir/NNMPerformanceSPI/bin/stopSPI.ovpl
$NnmInstallDir/NNMPerformanceSPI/bin/stopERS.ovpl
```

To prevent data corruption, ensure that no instance of the iSPI is running and accessing the shared disk.

3 (UNIX only) Activate the disk group:

```
vgchange -a e <disk_group>
```

4 Use the appropriate operating system commands to mount the shared disk. For example:

- *Windows*: Use Windows Explorer.
- *UNIX*: `mount /dev/vgnnm/lvnnm /nnmmount`

5 Copy the iSPI files from the shared disk to the node:

- *Windows*:

```
%NnmInstallDir%\misc\nnm\ha\nnmhadisk.ovpl PerfSPIHA \
-from <HA_mount_point>
```

- *UNIX*:

```
$NnmInstallDir/misc/nnm/ha/nnmhadisk.ovpl PerfSPIHA \
-from <HA_mount_point>
```

6 Use the appropriate operating system commands to unmount the shared disk. For example:

- *Windows*: Use Windows Explorer and Disk Management.
- *UNIX*: `umount /nnmmount`

7 (UNIX only) Deactivate the disk group:

```
vgchange -a n <disk_group>
```

8 Start the NNM iSPI for Performance:

- *Windows*:

```
%NnmInstallDir%\NNMPerformanceSPI\bin\startSPI.ovpl
%NnmInstallDir%\NNMPerformanceSPI\bin\startERS.ovpl
```

- *UNIX*:

```
$NnmInstallDir/NNMPerformanceSPI/bin/startSPI.ovpl
$NnmInstallDir/NNMPerformanceSPI/bin/startERS.ovpl
```

The NNM iSPI for Performance is now running with a copy of the database that was formerly used by the performance HA resource group.

Upgrading NNMi under HA from NNMi 8.0x to NNMi 8.11

To upgrade from NNMi 8.0x under HA to NNMi 8.11 under HA, unconfigure NNMi 8.0x from HA, install NNMi 8.11 to get the updated database schema, and then reconfigure NNMi for HA. Follow these steps:

1 Verify that the NNMi 8.0x configuration is consistent across all HA nodes:

- *Windows*:

```
%NnmInstallDir%\misc\nnm\ha\nnmdatareplicator.ovpl NNM
```

- *UNIX*:

```
$NnmInstallDir/misc/nnm/ha/nnmdatareplicator.ovpl NNM
```

2 Determine which node in the NNMi 8.0x HA cluster is active:

- *Windows*:

```
%NnmInstallDir%\misc\nnm\ha\nnmhaclusterinfo.ovpl \
-group <resource_group> -state
```

- *UNIX*:

```
$NnmInstallDir/misc/nnm/ha/nnmhaclusterinfo.ovpl \
-group <resource_group> -state
```

3 On each passive node, unconfigure NNMi 8.0x:

a Unconfigure NNMi from the HA cluster:

— *Windows*:

```
%NnmInstallDir%\misc\nnm\ha\nnmhaunconfigure.ovpl \
<resource_group>
```

— *UNIX*:

```
$NnmInstallDir/misc/nnm/ha/nnmhaunconfigure.ovpl \
<resource_group>
```

This command removes access to the shared disk but does not unconfigure the disk group or the volume group.

b Remove the resource group-specific files:

— *Windows*:

In Windows Explorer, delete all files in the `%NnmDataDir%\hacluster\<resource_group>\` folder.

— *HP-UX*:

```
rm -rf $NnmDataDir/hacluster/<resource_group>/*
rm -rf /etc/cmcluster/<resource_group>/*
```

— *Linux*:

```
rm -rf $NnmDataDir/hacluster/<resource_group>/*
rm -rf /usr/local/cmcluster/conf/<resource_group>/*
```

— *Solaris*:

```
rm -rf $NnmDataDir/hacluster/<resource_group>/*
```

4 On the active node, unconfigure NNMi 8.0x:

a Disable HA resource group monitoring by creating the following maintenance file:

— *Windows*: `%NnmDataDir%\hacluster\maint_NNM`

— *UNIX*: `$NnmDataDir/hacluster/<resource_group>/maint_NNM`

The file can be empty.

b Stop NNMi:

```
ovstop -c
```

To prevent data corruption, ensure that no instance of NNMi is running and accessing the shared disk.

c Back up the NNMi database that is on the shared disk:

```
nnmbackup.ovpl -type offline -target <backup_directory>
```

For more information on this command, see NNMi Backup and Restore Tools on page 181.

d Copy the NNMi files from the shared disk to the node:

— *Windows*:

```
%NnmInstallDir%\misc\nnm\ha\nnmhadisk.ovpl \
-from <HA_mount_point>
```

— *UNIX*:

```
$NnmInstallDir/misc/nnm/ha/nnmhadisk.ovpl \
-from <HA_mount_point>
```

e Remove all NNMi files and directories from the shared disk:

— *Windows*: Use Windows Explorer to delete all files under the shared disk mount point (`%HA_MOUNT_POINT%`, which is, for example `S:\`).

— *UNIX*:

```
rm -rf $HA_MOUNT_POINT/*
```

f Stop the NNMi HA resource group:

— *Windows*:

```
%NnmInstallDir%\misc\nnm\ha\nnmhastoprg.ovpl NNM \
<resource_group>
```

— *UNIX*:

```
$NnmInstallDir/misc/nnm/ha/nnmhastoprg.ovpl NNM \
<resource_group>
```

g Unconfigure NNMi from the HA cluster:

— *Windows*:

```
%NnmInstallDir%\misc\nnm\ha\nnmhaunconfigure.ovpl \
<resource_group>
```

— *UNIX*:

```
$NnmInstallDir/misc/nnm/ha/nnmhaunconfigure.ovpl \
<resource_group>
```

This command removes access to the shared disk but does not unconfigure the disk group or the volume group.

h Remove the resource group-specific files:

— *Windows*:

In Windows Explorer, delete all files in the `%NnmDataDir%\hacluster\<resource_group>\` folder.

— *HP-UX*:

```
rm -rf $NnmDataDir/hacluster/<resource_group>/*
rm -rf /etc/cmcluster/<resource_group>/*
```

— *Linux*:

```
rm -rf $NnmDataDir/hacluster/<resource_group>/*
rm -rf /usr/local/cmcluster/conf/<resource_group>/*
```

— *Solaris*:

```
rm -rf $NnmDataDir/hacluster/<resource_group>/*
```

i Delete the maintenance file:

— *Windows*: `%NnmDataDir%\hacluster\maint_NNM`

— *UNIX*: `$NnmDataDir/hacluster/<resource_group>/maint_NNM`

5 On the node that was active before unconfiguring NNMi 8.0x from HA, configure NNMi 8.11:

a Install the NNMi 8.11 upgrade.

The upgrade process starts NNMi to update the NNMi database schema.

b Install the latest NNMi consolidated patch (if any).

c Follow the steps in Configuring NNMi on the Primary Cluster Node on page 135.

You do not need to do the following:

— Define a disk device group and logical volume.

— Create a mount point for the shared disk.

— Configure the shared disk.

6 On each node that was passive before unconfiguring NNMi 8.0x from HA, configure NNMi 8.11:

a Install the NNMi 8.11 upgrade.

b Install the latest NNMi consolidated patch (if any).

c On the first secondary node, follow the steps in Configuring NNMi on the Secondary Cluster Nodes on page 138.

You do not need to create a mount point for the shared disk.

Troubleshooting the HA Configuration

General HA Troubleshooting

The topics in this section apply to HA configuration for all HP NNM i-series Software products.

Error: Wrong Number of Arguments

As of NNMi 8.10, the name of the product Perl module is a required parameter to most of the NNMi HA configuration commands.

- For NNMi use the value `NNM`.
- To determine what value to use for an iSPI, see Table 12 on page 147 or the appropriate iSPI chapter in the Integrations and Plug-ins section.

Product Startup Times Out (Solaris)

One or more of the `/var/adm/messages*` files contains a message similar to the following example:

```
VCS ERROR V-16-1-13012 Thread(…) Resource(<resource group>-app):
online procedure did not complete within the expected time.
```

This message indicates that the product did not start completely within the Veritas timeout value. The NNMi-provided HA configuration scripts define this timeout to be 15 minutes.

To change the Veritas timeout value on the Solaris operating system, run the following commands in order:

```
/opt/VRTSvcs/bin/haconf -makerw
/opt/VRTSvcs/bin/hares -modify <resource_group>-app OnlineTimeout <value in seconds>
/opt/VRTSvcs/bin/haconf -dump -makero
```

Log Files on the Active Cluster Node Are Not Updating

This situation is normal. It occurs because the log files have been redirected to the shared disk.

For NNMi, review the log files in the location specified by `HA_NNM_LOG_DIR` in the `ov.conf` file.

For the NNM iSPI for Performance, the local log files are removed during HA configuration. For the location of the iSPI log files on the shared disk, see NNM iSPI for Performance HA Log Files on page 180.

Cannot Start the HA Resource Group on a Particular Cluster Node

If the `nnmhargconfigure.ovpl` command does not correctly start, stop, or switch the NNMi HA resource group, review the following information:

- *Windows*:
  - In Cluster Administrator, review the state of the resource group and underlying resources.
  - Review the Event Viewer log for any errors.
- *HP-UX*:

  Review the `/etc/cmcluster/<resource_group>/<resource_group>.cntl.log` file and the syslog files for errors. The most common problems are leaving the system in a state where a resource cannot be added, for example, having a disk group misconfigured such that it cannot be activated.
- *Linux*:

  Review the `/usr/local/cmcluster/conf/<resource_group>/<resource_group>.cntl.log` file and the syslog files for errors. The most common problems are leaving the system in a state where a resource cannot be added, for example, having a disk group misconfigured such that it cannot be activated.
- *Solaris*:
  - Run **`/opt/VRTSvcs/bin/hares -state`** to review the resource state.
  - For failed resources, review the `/var/VRTSvcs/log/<resource>.log` file for the resource that is failing. Resources are referenced by the agent type, for example: `IP*.log`, `Mount*.log`, and `Volume*.log`.

If you cannot locate the source of the problem, you can manually start the HA resource group by using the HA product commands:

1. Mount the shared disk.
2. Assign the virtual host to the network interface:
   - *Windows*:
     - Start Cluster Administrator.
     - Expand the resource group.
     - Right-click ***<resource_group>*-ip**, and then click **Bring Online**.
   - *HP-UX*: Run `/usr/sbin/cmmodnet` to add the IP address.
   - *Linux*: Run `/usr/local/cmcluster/bin/cmmodnet` to add the IP address.
   - *Solaris*: **`/opt/VRTSvcs/bin/hares -online <resource_group>-ip \ -sys <local_hostname>`**

3 Start the HA resource group. For example:

- *Windows*:

```
%NnmInstallDir%\misc\nnm\ha\nnmhastart.ovpl NNM \
-start <resource_group>
```

- *UNIX*:

```
$NnmInstallDir/misc/nnm/ha/nnmhastart.ovpl NNM \
-start <resource_group>
```

The return code `0` indicates that NNMi started successfully.

The return code `1` indicates that NNMi did not start correctly.

NNMi-Specific HA Troubleshooting

The topics in this section apply to HA configuration for NNMi only.

Re-Enable NNMi for HA after All Cluster Nodes are Unconfigured

When all NNMi HA cluster nodes have been unconfigured, the `ov.conf` file no longer contains any mount point references to the NNMi shared disk. To re-create the mount point reference without overwriting the data on the shared disk, follow these steps on the primary node:

1 If NNMi is running, stop it:

```
ovstop -c
```

2 Reset the reference to the shared disk:

- *Windows*:

```
%NnmInstallDir%\misc\nnm\ha\nnmhadisk.ovpl NNM \
-setmount <HA_mount_point>
```

- *UNIX*:

```
$NnmInstallDir/misc/nnm/ha/nnmhadisk.ovpl NNM \
-setmount <HA_mount_point>
```

3 In the `ov.conf` file, verify the entries related to HA mount points.

For the location of the `ov.conf` file, see NNMi HA Configuration Files on page 177.

NNMi Does Not Start Correctly Under HA

Either of the following situations occurs:

- `ovspmd` does not start after NNMi is configured for HA. (`ovstatus` shows that `ovspmd` is NOT RUNNING.)
- `ovstart` or `ovstop` returns the following message:

```
ovstart: Must specify managers by name if running on OVW
Client
```

While NNMi is running under HA, *do not* use the `ovstart` and `ovstop` commands unless instructed to do so for HA maintenance purposes.

The NNMi configuration points to a different system than where NNMi is running. To fix the problem, verify that the `ov.conf` file has appropriate entries for the following items:

- `NNM_INTERFACE=<virtual_hostname>`
- `NNM_HA_CONFIGURED=YES`
- `HA_RESOURCE_GROUP=<resource_group>`
- `HA_POSTGRES_DIR=<HA_mount_point>/NNM/dataDir/shared/nnm/databases/Postgres`
- `HA_EVENTDB_DIR=<HA_mount_point>/NNM/dataDir/shared/nnm/eventdb`
- `HA_NNM_LOG_DIR=<HA_mount_point>/NNM/dataDir/log`
- `HA_JBOSS_DATA_DIR=<HA_mount_point>/NNM/installDir/nonOV/jboss/nms/server/nms/data`
- `HA_MOUNT_POINT=<HA_mount_point>`

For the location of the `ov.conf` file, see NNMi HA Configuration Files on page 177.

nmsdbmgr Does Not Start after HA Configuration

This situation usually occurs as a result of starting NNMi immediately after running the `nnmhadisk.ovpl` command with the `-to` option. (The `nnmhaconfigure.ovpl` command has not been run.) In this case, the `HA_POSTGRES_DIR` entry in the `ov.conf` file specifies the location of the embedded database on the shared disk, but this location is not available to NNMi.

Run `nnmhaconfigure.ovpl`, and then complete the HA configuration.

pmd Does Not Start after HA Configuration

This situation usually occurs after a configuration error such as not setting up the shared disk correctly. The failure of the `pmd` process occurs when the `ovjboss` process does not fully start.

Review the following log file:

- *Windows*: `%HA_MOUNT_POINT%\NNM\dataDir\log\nnm\jbossServer.log`
- *UNIX*: `$HA_MOUNT_POINT/NNM/dataDir/log/nnm/jbossServer.log`

Disk Failover Does Not Occur

This situation can happen when the shared disk is not supported for the appropriate platform. Review the appropriate HA product, operating system, and disk manufacturer documentation to determine whether these products can all work together.

If disk failure occurs, NNMi will not start upon failover. Most likely, `nmsdbmgr` will fail because the `HA_POSTGRES_DIR` directory does not exist. Verify that the shared disk is mounted and that the appropriate files are accessible.

Shared Disk Files Are Not Found on the Secondary Node after Failover

The most common cause of this situation is that the `nnmhadisk.ovpl` command was run with the `-to` option when the shared disk was not mounted. In this case, the data files are copied to the local disk, so the files are not available on the shared disk.

iSPI-Specific HA Troubleshooting

The topics in this section apply to add-on HA configuration for all HP NNM i-series Software iSPIs.

iSPIs Do Not Start Correctly Under HA

Call HP Support for assistance before you begin this procedure.

1. On the cluster node in question, determine whether each iSPI is correctly registered by examining the current values for the `NNM_ADD_ON_PRODUCTS` attribute:
   - *Windows*: Run the following command:

     `cluster group <resource_group> /PRIV`

     In the output, look for `PUBLIC.NNM_ADD_ON_PRODUCTS`.
   - *HP-UX*: Look in the `/etc/cmcluster/<resource_group>/<resource_group>.public.env` file.
   - *Linux*: Look in the `/usr/local/cmcluster/conf/<resource_group>/<resource_group>.public.env` file.
   - *Solaris*: Stored as part of the resource group specification in `UserStrGlobal`.
2. If the values for `NNM_ADD_ON_PRODUCTS` are not correct, update the values:
   - *Windows*: Run the following command:

     `cluster group <resource_group> /PRIV \`
     `<attribute_name>=<value>`

     Do not edit the Windows Registry directly.
   - *HP-UX*: Copy a good version of the `/etc/cmcluster/<resource_group>/<resource_group>.public.env` file from another cluster node to this one.
   - *Linux*: Copy a good version of the `/usr/local/cmcluster/conf/<resource_group>/<resource_group>.public.env` file from another cluster node to this one.
   - *Solaris*: Update `UserStrGlobal`:

     `/opt/VRTSvcs/bin/haconf -makerw`
     `/opt/VRTSvcs/bin/hagrp -modify <resource_group> \`
     `UserStrGlobal "<newvalue>"`

HA Configuration Reference

NNMi HA Configuration Files

Table 14 lists the NNMi HA configuration files. These files apply to NNMi and add-on iSPIs on the NNMi management server. These files are installed to the following location:

- *Windows*: `%NnmDataDir%\shared\nnm\conf`
- *UNIX*: `$NnmDataDir/shared/nnm/conf`

Table 14 NNMi HA Configuration Files

File Name	Description
ov.conf	Updated by the `nnmhaclusterinfo.ovpl` command to describe the NNMi HA implementation. NNMi processes read this file to determine the HA configuration.
nnmdatareplicator.conf	Used by the `nnmdatareplicator.ovpl` command to determine which NNMi folders and files are included in data replication from the active node to the passive nodes. If you implement a different method of replicating the NNMi configuration, see this file for a list of the data to include. For more information, see the comments in the file.

NNM iSPI for Performance HA Configuration Files

The add-on NNM iSPI for Performance uses the `ov.conf` file for configuring HA.

The NNM iSPI for Performance on a dedicated server uses the following file for configuring HA:

- *Windows*: `%NnmDataDir%\shared\perfSpi\conf\perfspi.conf`
- *UNIX*: `$NnmDataDir/shared/perfSpi/conf/perfspi.conf`

NNMi-Provided HA Configuration Scripts

Table 15 lists the HA configuration scripts that are included with NNMi. The NNMi-provided scripts are convenience scripts that can be used to configure HA for any product that has a customer Perl module. If you prefer, you can use the HA product-provided commands to configure HA for NNMi.

On the NNMi management server, the NNMi-provided HA configuration scripts are installed to the following location:

- *Windows*: `%NnmInstallDir%\misc\nnm\ha`
- *UNIX*: `$NnmInstallDir/misc/nnm/ha`

On the HP-UX and Solaris operating systems, this directory contains obsoleted scripts (with names `ov*`). Do not run these scripts.

Table 15 NNMi HA Configuration Scripts

Script Name	Description
nnmhaconfigure.ovpl	Configures NNMi or an iSPI for an HA cluster. Run this script on all nodes in the HA cluster.
nnmhaunconfigure.ovpl	Unconfigures NNMi or an iSPI from an HA cluster. Optionally, run this script on one or more nodes in the HA cluster.
nnmhaclusterinfo.ovpl	Retrieves cluster information regarding NNMi. Run this script as needed on any node in the HA cluster.
nnmhadisk.ovpl	Copies NNMi and iSPI data files to and from the shared disk. During HA configuration, run this script on the primary node. At other times, run this script per the instructions in this chapter.
nnmhastartrg.ovpl	Starts NNMi in an HA cluster. During HA configuration, run this script on the primary node.
nnmdatareplicator.ovpl	Checks the `nnmdatareplicator.conf` configuration file for changes and copies files to remote systems.
nnmharg.ovpl	Starts, stops, and monitors NNMi in an HA cluster. Used by `<resource_group>.cntl` for ServiceGuard configurations (HP-UX and Linux). Used by the Veritas start, stop, and monitor scripts on Solaris. (`nnmhargconfigure.ovpl` configures this usage.) Also used by `nnmhastartrg.ovpl` to enable and disable tracing.
nnmhargconfigure.ovpl	Configures HA resources and resource groups. Used by `nnmhaconfigure.ovpl` and `nnmhaunconfigure.ovpl`.
nnmhastart.ovpl	Starts NNMi in an HA cluster. Used by `nnmharg.ovpl`.
nnmhastop.ovpl	Stops NNMi in an HA cluster. Used by `nnmharg.ovpl`.
nnmhamonitor.ovpl	Monitors NNMi processes in an HA cluster. Used by `nnmharg.ovpl`.
nnmhamscs.vbs	Is a template for creating a script to start, stop, and monitor NNMi processes in a Microsoft Cluster Services HA cluster. The generated script is used by Microsoft Cluster Services and is stored in the following location: `%NnmDataDir%\hacluster\<resource_group>\hamscs.vbs`

NNMi HA Configuration Log Files

The following log files apply to the HA configuration for NNMi and add-on iSPIs on the NNMi management server:

- *Windows* configuration:
  - `%NnmDataDir%\tmp\HA_nnmhaserver.log`
  - `%NnmDataDir%\log\haconfigure.log`
- *UNIX* configuration:
  - `$NnmDataDir/tmp/HA_nnmhaserver.log`
  - `$NnmDataDir/log/haconfigure.log`
- *Windows* runtime:
  - Event Viewer log
  - `%HA_MOUNT_POINT%\NNM\dataDir\log\nnm\ovspmd.log`
  - `%HA_MOUNT_POINT%\NNM\dataDir\log\nnm\public\postgres.log`
  - `%HA_MOUNT_POINT%\NNM\dataDir\log\nnm\public\nmsdbmgr.log`
  - `%HA_MOUNT_POINT%\NNM\dataDir\log\nnm\jbossServer.log`
  - `%HA_MOUNT_POINT%\NNM\dataDir\log\nnm\ovet*.log`
  - `%SystemRoot%\Cluster\cluster.log`
    This is the log file for cluster runtime issues including: adding and removing resources and resource groups; other configuration issues; starting and stopping issues.
- *HP-UX* runtime:
  - `/etc/cmcluster/<resource_group>/<resource_group>.cntl.log`
    This is the log file for the resource group.
  - `/var/adm/syslog/syslog.log`
  - `/var/adm/syslog/OLDsyslog.log`
  - `$HA_MOUNT_POINT/NNM/dataDir/log/nnm/ovspmd.log`
  - `$HA_MOUNT_POINT/NNM/dataDir/log/nnm/public/postgres.log`
  - `$HA_MOUNT_POINT/NNM/dataDir/log/nnm/public/nmsdbmgr.log`
  - `$HA_MOUNT_POINT/NNM/dataDir/log/nnm/jbossServer.log`
  - `$HA_MOUNT_POINT/NNM/dataDir/log/nnm/ovet*.log`
- *Linux* runtime:
  - `/usr/local/cmcluster/conf/<resource_group>/<resource_group>.cntl.log`
    This is the log file for the resource group.
  - `/var/log/cmcluster`
    This is the log file for cluster issues.
  - `/var/log/messages*`
    These are the log files for resource issues (disk and IP address).
  - `$HA_MOUNT_POINT/NNM/dataDir/log/nnm/ovspmd.log`
  - `$HA_MOUNT_POINT/NNM/dataDir/log/nnm/public/postgres.log`

- — `$HA_MOUNT_POINT/NNM/dataDir/log/nnm/public/nmsdbmgr.log`
- — `$HA_MOUNT_POINT/NNM/dataDir/log/nnm/jbossServer.log`
- — `$HA_MOUNT_POINT/NNM/dataDir/log/nnm/ovet*.log`

- *Solaris* runtime:

Resource	Log File
<*resource_group*>-app	• `/var/VRTSvcs/log/Application_A.log` • `$HA_MOUNT_POINT/NNM/dataDir/log/nnm/ovspmd.log` • `$HA_MOUNT_POINT/NNM/dataDir/log/nnm/public/postgres.log` • `$HA_MOUNT_POINT/NNM/dataDir/log/nnm/public/nmsdbmgr.log` • `$HA_MOUNT_POINT/NNM/dataDir/log/nnm/jbossServer.log` • `$HA_MOUNT_POINT/NNM/dataDir/log/nnm/ovet*.log` • `/var/adm/messages*`
<*resource_group*>-dg <*resource_group*>-volume <*resource_group*>-mount	• `/var/VRTSvcs/log/DiskGroup_A.log` • `/var/VRTSvcs/log/Volume_A.log` • `/var/VRTSvcs/log/Mount_A.log` • `/var/adm/messages*`
<*resource_group*>-ip	• `/var/VRTSvcs/log/IP_A.log` • `/var/adm/messages*`
For operating system-specific issues related to the HA resources, review the `/var/adm/messages*` files. For <*resource_group*>-app, look for messages regarding unable to start process.	

NNM iSPI for Performance HA Log Files

The performance HA resource group uses many of the log files listed in NNMi HA Configuration Log Files on page 179. Additionally, the configuration maintains the following log files:

- *Windows* configuration:
  - — `%NnmDataDir%\NNMPerformanceSPI\logs\prspiHA.log`
- *UNIX* configuration:
  - — `$NnmDataDir/NNMPerformanceSPI/logs/prspiHA.log`
- *Windows* runtime:
  - — `%HA_MOUNT_POINT%\NNMPerformanceSPI\dataDir\NNMPerformanceSPI\logs\*.log`
- *UNIX* runtime:
  - — `$HA_MOUNT_POINT/NNMPerformanceSPI/dataDir/NNMPerformanceSPI/logs/*.log`

NNMi Backup and Restore Tools

This chapter contains the following topics:

About NNMi Data Backup

A good backup and restore strategy is key to ensuring the uninterrupted operations of any business. HP Network Node Manager i Software is an important asset for network operations and should be backed up regularly.

HP recommends that you perform a complete weekly backup of all NNMi data. You do not need to shut down NNMi in order to create this backup. Perform scoped backups (as described in Backing up NNMi Data) as needed before beginning configuration changes. In this way, if your configuration changes do not have the expected effect, you will be able to revert to a known working configuration.

The two types of critical data related to an NNMi installation are as follows:

- Files in the file system
- Data in the relational database (embedded or external)

This chapter explains the tools that NNMi provides for backing up and restoring important NNMi files and data. If you use the NNMi embedded database, these tools might be sufficient for backing up and restoring the NNMi installation. If you use an external database (such as Oracle), read this document to identify the NNMi files and database tables that should be added to your existing database backup procedures.

Script names

NNMi provides the following scripts for backing up and restoring NNMi data:

- `nnmbackup.ovpl`—Backs up all necessary file system data (including configuration information) and any data stored in the embedded database.
- `nnmrestore.ovpl`—Restores a backup that was created by using the `nnmbackup.ovpl` script.
- `nnmbackupembdb.ovpl`—Creates a complete backup of the NNMi embedded database (but not the file system data) while NNMi is running.
- `nnmrestoreembdb.ovpl`—Restores a backup that was created by using the `nnmbackupembdb.ovpl` script.

Interaction with iSPIs

Data from Smart Plug-ins (iSPIs) can be included in the NNMi backups and restores. For information about a particular iSPI, see the appropriate chapter in the Integrations and Plug-ins section of this guide.

Limitations of the NNMi Backup and Restore Scripts

Backup and Restore Systems Must Be of the Same Configuration

You can use the backup and restore scripts to transfer data from one NNMi management server to another. The following items must be identical on both systems:

- NNMi version (including any patches)
- Operating system type and version
- Character set (language)

The following items can differ between the two systems:

- Hostname
- IP address

External Databases Are Not Supported

The NNMi backup and restore scripts do not work with data stored in an external database (such as Oracle). External database maintenance should be handled as part of the existing database backup and restore procedures.

Backing up NNMi Data

The NNMi-provided backup script copies the synchronized NNMi data to the specified target directory. You can then use any appropriate tool to save a copy of the backup.

The NNMi backup script supports two types of backups:

- Online backups occur while NNMi is running. At a minimum, the nmsdbmgr service must be running.
- Offline backups occur while NNMi is completely stopped.

You can use the NNMi backup script to create a complete backup of the NNMi data or to back up only some of the data according to function. A partial backup is useful for moving information between systems and for archiving the current state of NNMi.

Partial backups are only available for online backups. If you run the backup script while NNMi is completely stopped, the script copies all data in the embedded database regardless of the backup scope.

The syntax for the backup script command is as follows. Table 16 describes the script command line options.

```
nnmbackup.ovpl [-type (online|offline)]
   [-scope (config|topology|events|all)] [-force] [-archive]
   -target <directory>]
```

Table 16 nnmbackup.ovpl Command Line Options

Option	Description
-type (online | offline)	Specifies the type of backup. For an offline backup, stop NNMi prior to running the command or specify the `-force` option. If no value is specified, an online backup is performed.
-scope (config | topology | events | all)	Specifies the data to back up. • For file system data, the scope applies to both online and offline backups. • For database tables, the scope applies to online backups only. For offline backups, the entire embedded database is backed up regardless of the scope. If no value is specified, a complete backup is performed.
-force	Forces NNMi into the appropriate state for the type of backup. For an online backup, the command starts the nmsdbmgr service if it is not already running. For an offline backup, the command stops all NNMi services. If this option is not specified and NNMi is not in the correct state for the specified backup type, the script fails.
-archive	Creates an archive file (tar file) in the target directory.
-target <*directory*>	Specifies the directory to contain the backup files.

Configuration scope

The configuration scope (`-scope config`) includes the files and database tables that store NNMi configuration information only. The configuration scope loosely aligns to the information in the **Configuration** workspace of the NNMi console.

Table 17 identifies the files and directories that are backed up for the configuration scope.

Table 17 Configuration Files and Directories

Directory or File name	Description
$NnmInstallDir/conf (Windows only)	Configuration information
$NnmDataDir/conf	Configuration that might be shared by other HP products
$NnmDataDir/shared/nnm/conf	Shared configuration data within NNMi

Table 17 Configuration Files and Directories (cont'd)

Directory or File name	Description
$NnmDataDir/shared/nnm/lrf	Shared configuration data within NNMi— component registration files
$NnmDataDir/shared/nnm/databases/Postgres	File system storage for the NNMi embedded database **NOTE**: This directory is only backed up for offline backups.
$NnmDataDir/NNMVersionInfo	NNMi version information file
$NnmInstallDir/misc/nnm	Miscellaneous configuration data
$NnmInstallDir/newconfig	Installation configuration staging area
$NnmInstallDir/nonOV/jboss/nms/server/nms/conf	jboss configuration
$NnmInstallDir/nonOV/jboss/nms/server/nms/deploy	jboss deployment directory
$NnmInstallDir/snmp_mibs (Windows only) $NnmDataDir/share/snmp_mibs (UNIX only)	SNMP MIB information
$NnmDataDir/HPOvLIC/LicFile.txt	License information

Topology scope

The topology scope (`-scope topology`) includes the files and database tables that store NNMi network topology information. The topology scope loosely aligns to the information in the **Inventory** workspace of the NNMi console.

Because the network topology is dependent on the configuration that was used for discovering that topology, the topology scope includes the configuration scope.

Event scope

The event scope (`-scope event`) includes the files and database tables that store NNMi network incident and event information. The event scope loosely aligns to the information in the **Incident Browsing** workspace of the NNMi console.

Because events are dependent on the network topology related to those events, the event scope includes the configuration and topology scopes.

Complete backup

The complete backup (`-scope all`) includes all important NNMi files and data. It is the conjunction of the configuration, topology, and event scopes.

The Custom Poller database tables are currently included in the complete backup only.

Restoring NNMi Data

The NNMi restore script copies the data from the specified source directory to the appropriate locations in the file system and the embedded database. You can restore data from a partial backup or from a complete backup.

- The NNMi restore script only works with the information collected by the NNMi backup script. Copy the saved backup data to a directory that is accessible from the NNMi management server, and then run the restore script to copy that data to the correct locations.
- Because the database schema might change from one version of NNMi to the next, data backups cannot be shared across versions of NNMi.

Stop all NNMi services before running the restore script. If you are restoring data from an online backup, use the `-force` option

During a restoration, the target database is completely cleaned out and replaced with the contents of the backup. This cleaning can be useful for things like moving configuration information from one system to another (perhaps because of new hardware).

If the provided source is a tar file, the restore command extracts the tar file to a temporary folder in the current working directory. In this case, either ensure that the current working directory has adequate storage to support the temporary folder, or extract the archive before running the restore command.

- If you are restoring a backup from a different NNMi management server, check the license information from the NNMi console login page to determine whether the license information is correct for the new system. If the restored license is not valid for the new NNMi management server, obtain and apply a new license for the new NNMi management server. For more information, see the licensing documentation.

The syntax for the restore script command is as follows. Table 18 describes the script command line options.

```
nnmrestore.ovpl [-force] -source <directory>
```

Table 18 nnmrestore.ovpl Command Line Options

Option	Description
-force	Forces NNMi into the appropriate state for the type of restoration. For all restorations, the command stops all NNMi services. For a restoration of an online backup, the command starts the nmsdbmgr service at the appropriate point in the restoration process.
-source <*directory*>	Specifies the directory that contains the backup files.

Backing up and Restoring the Embedded Database Only

NNMi provides scripts to backup and restore the embedded database only, which is useful for creating a snapshot of the data as you experiment with NNMi configuration settings. The embedded database backup and restore scripts perform online backups only. At a minimum, the nmsdbmgr service must be running.

Best practice

Run the embedded database reset script (`nnmresetembdb.ovpl`) before restoring data to the embedded database. This script ensures that the database does not contain any errors, thereby eliminating the possibility of encountering database constraint violations. For information about running the embedded database reset script, see the `nnmresetembdb.ovpl` reference page, or the UNIX manpage.

The syntax for the embedded database backup script command is as follows. Table 19 describes the script command line options.

```
nnmbackupembdb.ovpl [-force] -target <file>
```

Table 19 nnmbackupembdb.ovpl Command Line Options

Option	Description
-force	Starts the nmsdbmgr service if it is not already running. If this option is not specified and the nmsdbmgr service is not running, the script fails.
-target <*file*>	Specifies the name of the backup file.

The syntax for the embedded database restore script command is as follows. Table 20 describes the script command line options.

```
nnmrestoreembdb.ovpl [-force] -source <file>
```

Table 20 nnmrestoreembdb.ovpl Command Line Options

Option	Description
-force	Starts the nmsdbmgr service if it is not already running. If this option is not specified and the nmsdbmgr service is not running, the script fails.
-source <*file*>	Specifies the name of the file that contains the backup.

Changing the NNMi Management Server

You can duplicate the HP Network Node Manager i Software configuration on another system, for example, to move from a test environment to a production environment or to change the hardware of the NNMi management server.

You can change the IP address of the NNMi management server without affecting the NNMi configuration.

This chapter contains the following topics:

Best Practices for Preparing the NNMi Configuration to Be Moved

The following best practices apply to moving the NNMi configuration to a different system:

- If the node group configuration uses hostnames to identify managed nodes, the production and test NNMi management servers must use the same DNS servers. In the case that the production and test systems use different DNS servers, changes in the resolved name for a managed node might result in different polling settings between the two NNMi management servers.
- You can limit the configuration export to a single author. Create a new author value that is unique to your group or company. Specify this author value when you create or modify any of the following items:
  - Device profile
  - Incident configuration
  - URL action
- If you plan to install Smart Plug-ins (iSPIs), see the appropriate chapters in the Integrations and Plug-ins section.

Moving the NNMi Configuration

Use the `nnmconfigexport.ovpl` command to output the NNMi configuration to an XML file. Then, use the `nnmconfigimport.ovpl` command to pull this configuration from the XML file into NNMi on the new system.

For information about these commands, refer to the appropriate reference pages, or the UNIX manpages.

You can only move the NNMi configuration. HP does not support moving topology or incident data from one NNMi management server to a different NNMi management server. Nor does HP support moving iSPI data, such as performance data that was collected for the NNM iSPI for Performance.

Changing the IP Address of an NNMi Management Server

If you need to change the IP address of the NNMi management server, follow these steps:

1. Go to **http://www.webware.hp.com**.
2. Click **Manage Licenses**.
3. Log in, and then follow the procedures to complete the move process to obtain your new license key.
4. Configure the NNMi management server with the new IP address.
5. Configure your DNS servers to recognize the new IP address of the NNMi management server.
6. Reboot the NNMi management server.
7. At a command prompt, enter the following command:

   ```
   nnmlicense.ovpl NNM -g
   ```

8. In the **Autopass: License Management** dialog box, click **Remove License Key**.
9. Select the license key to remove.
10. Select **Remove Licenses permanently.**
11. Click **Remove**, and then close the dialog box.
12. Copy the new license key that you obtained in step 3 into a text file named `license.txt`.
13. Run the following command:

    ```
    nnmlicense.ovpl NNM -f license.txt
    ```

Migration from NNM 6.x/7.x

This section contains the following chapters:

Product Comparison

This chapter describes the key differences between HP Network Node Manager (NNM) 6.x/7.x and HP Network Node Manager i Software. If you are familiar with a prior version of NNM, refer to this chapter as you plan and configure NNMi. If you are new to NNMi, you do not need to read this chapter.

This chapter contains the following topics:

Network Discovery

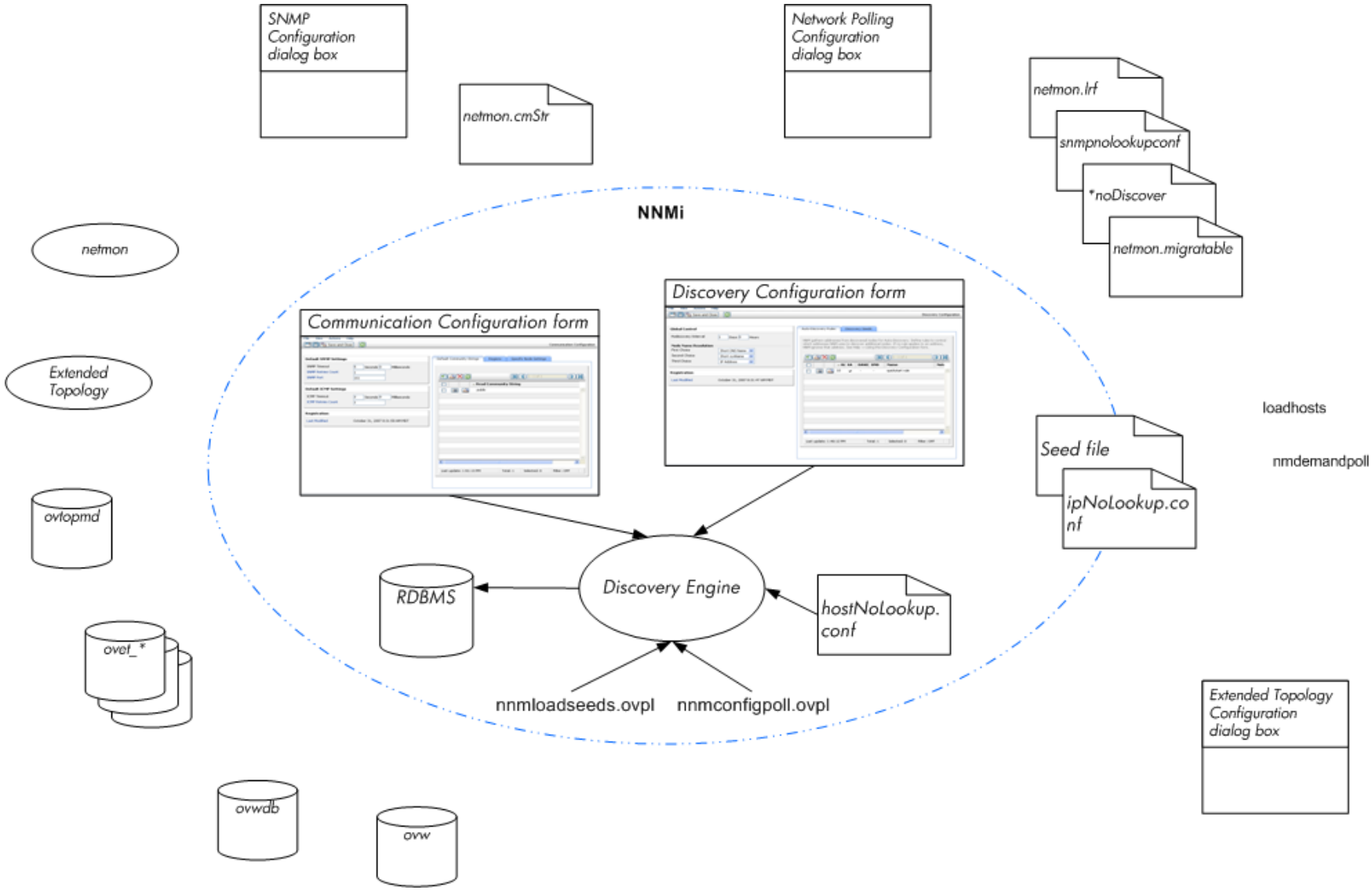
Discovery controls the network elements (devices, nodes, and their components) that are added into the database. In NNMi, “inventory discovery” refers to the activity of finding new nodes and “Layer 2 discovery” refers to the connectivity modeling previously performed by Extended Topology discovery.

In NNM 6.x/7.x, by default, when NNM started, it used its own loopback address as a seed and started automatic discovery of the network to which it was directly connected (based on its own IP address and subnet mask). NNMi allows the administrator control from the beginning. For NNMi auto-discovery, you define discovery regions based on IP address range(s) and specify at least one seed device (usually a router) before any discovery takes place.

The center of Figure 14 shows the tools, files, and commands used to configure discovery in NNMi. The perimeter of the figure shows the similar items for NNM 6.x/7.x.

The Extended Topology information applies to NNM 6.x with the Extended Topology add-on or NNM 7.x Advanced Edition only.

Figure 14 Discovery Configuration Elements



Key Concepts for Discovery

This section briefly describes the main areas of change from NNM 6.x/7.x to NNMi. For more information about NNMi discovery, refer to *Discovering Your Network* in the NNMi help.

- NNMi stores all information in one relational database.
- NNMi uses one consolidated discovery engine that is easy to configure.
- The NNMi Spiral Discovery process provides ongoing updates to the topology information as changes occur in your network. Topology changes (both inventory and Layer 2) can be discovered more frequently than the scheduled rediscovery interval.
- In NNMi, all discovered nodes are counted against the license limit regardless of the management mode (MANAGED, UNMANAGED, or OUT OF SERVICE). You cannot discover nodes beyond the license limit.
- Auto-discovery has the same meaning in NNMi as in NNM 6.x/7.x, but the configuration approach is different.
  - In NNMi, you define the auto-discovery boundaries, provide at least one IP address seed, and then let discovery run.
  - NNMi auto-discovery uses an expanding model that is easy to control. NNMi auto-discovery finds and manages all routers, switches, and subnets within the boundaries that you provide. You specify the additional device types that NNMi should discover and manage.

    By default, non-SNMP nodes are *not* discovered in NNMi.

- Seeded discovery has the same meaning in NNMi as in NNM 6.x/7.x, but the configuration approach is different.
  - In NNMi, you can specify discovery seeds in the user interface.
  - You can use your NNM 6.x/7.x seed files in NNMi without modification.
  - The NNMi `nnmloadseeds.ovpl` command replaces the NNM 6.x/7.x `loadhosts` command.
- The NNMi configuration poll (`nnmconfigpoll.ovpl`) replaces the NNM 6.x/7.x demand poll (`nmdemandpoll`) for determining device configuration information.

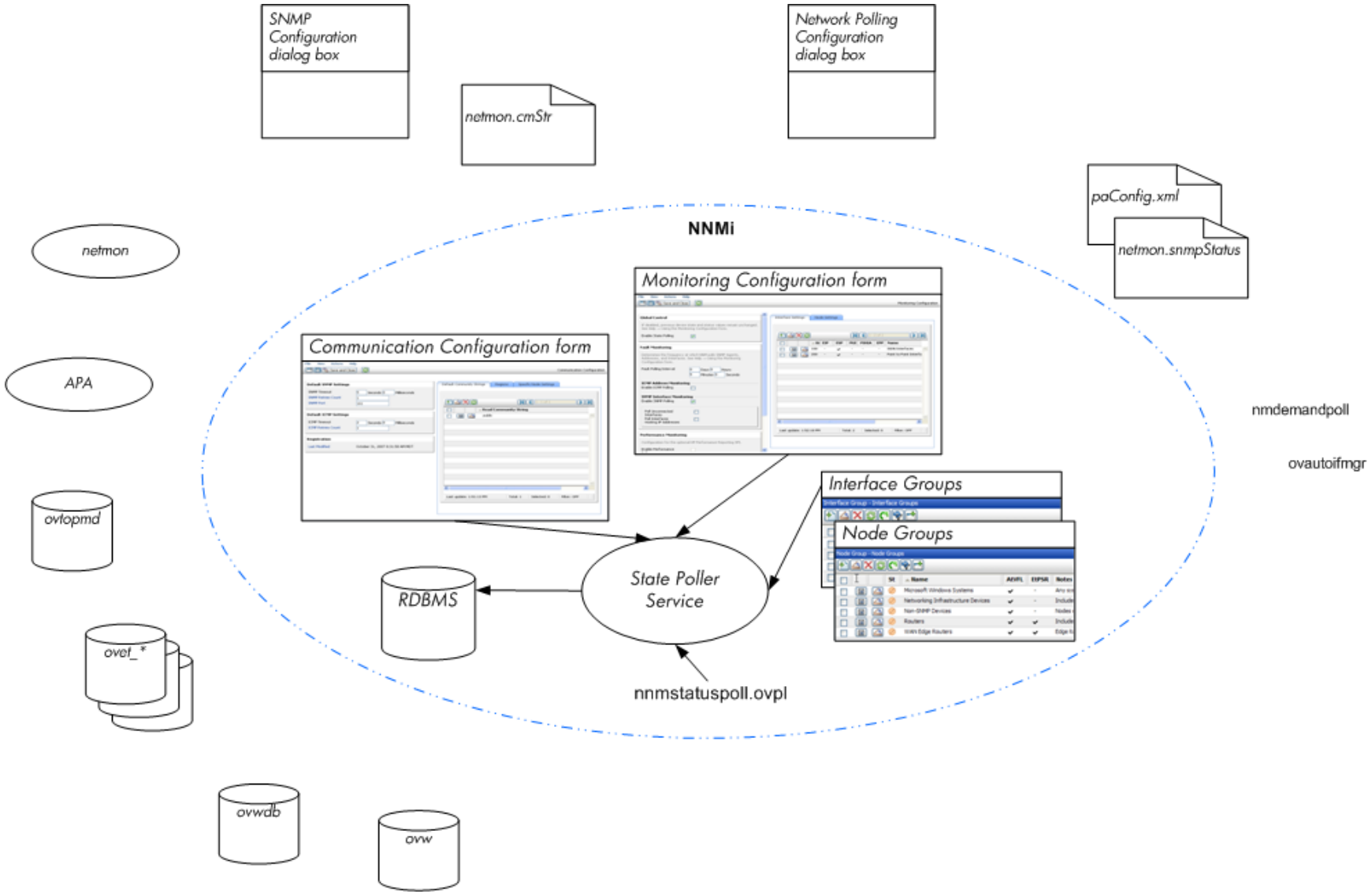
Status Monitoring

Status monitoring ensures that your network visualization is up-to-date in terms of devices or components that may have faults. When an element fails a poll, NNMi investigates the cause and issues a root cause alarm to the Incident Browser.

The center of Figure 15 shows the tools, files, and commands used to configure status monitoring in NNMi. The perimeter of the figure shows the similar items for NNM 6.x/7.x.

The APA information applies to NNM 7.x Advanced Edition only.

Figure 15 Monitoring Configuration Elements



Key Concepts for Status Monitoring

This section briefly describes the main areas of change from NNM 6.x/7.x to NNMi. For more information about NNMi status monitoring, refer to *Monitoring Network Health* in the NNMi help.

- Configuration is now completed through the user interface.
- NNMi node groups and interface groups replace topology filters.
  - Groups can be filtered on a pre-defined set of attributes only.
  - Groups cannot be connected with Boolean operators.
  - Node groups use device filters instead of relying on sysObjectId wildcards.
  - Interface groups can be restricted based on the group of the containing node and the interface type.
- ICMP polling is disabled by default.
- Broad controls make it easier to exclude uninteresting interfaces.
- Monitoring settings are matched from most specific to most general: (1) interface settings, (2) node settings, (3) defaults.
- To change a monitoring behavior across the system, change all settings at all levels.
- The NNMi status poll (**Actions > Status Poll** or `nnmstatuspoll.ovpl`) replaces the NNM 6.x/7.x demand poll (`nmdemandpoll`) for determining device status.
- By default, NNMi only polls interfaces that are connected to another known interface through a Layer 2 connection. You can enable polling of unconnected interfaces and of interfaces that host IP addresses.

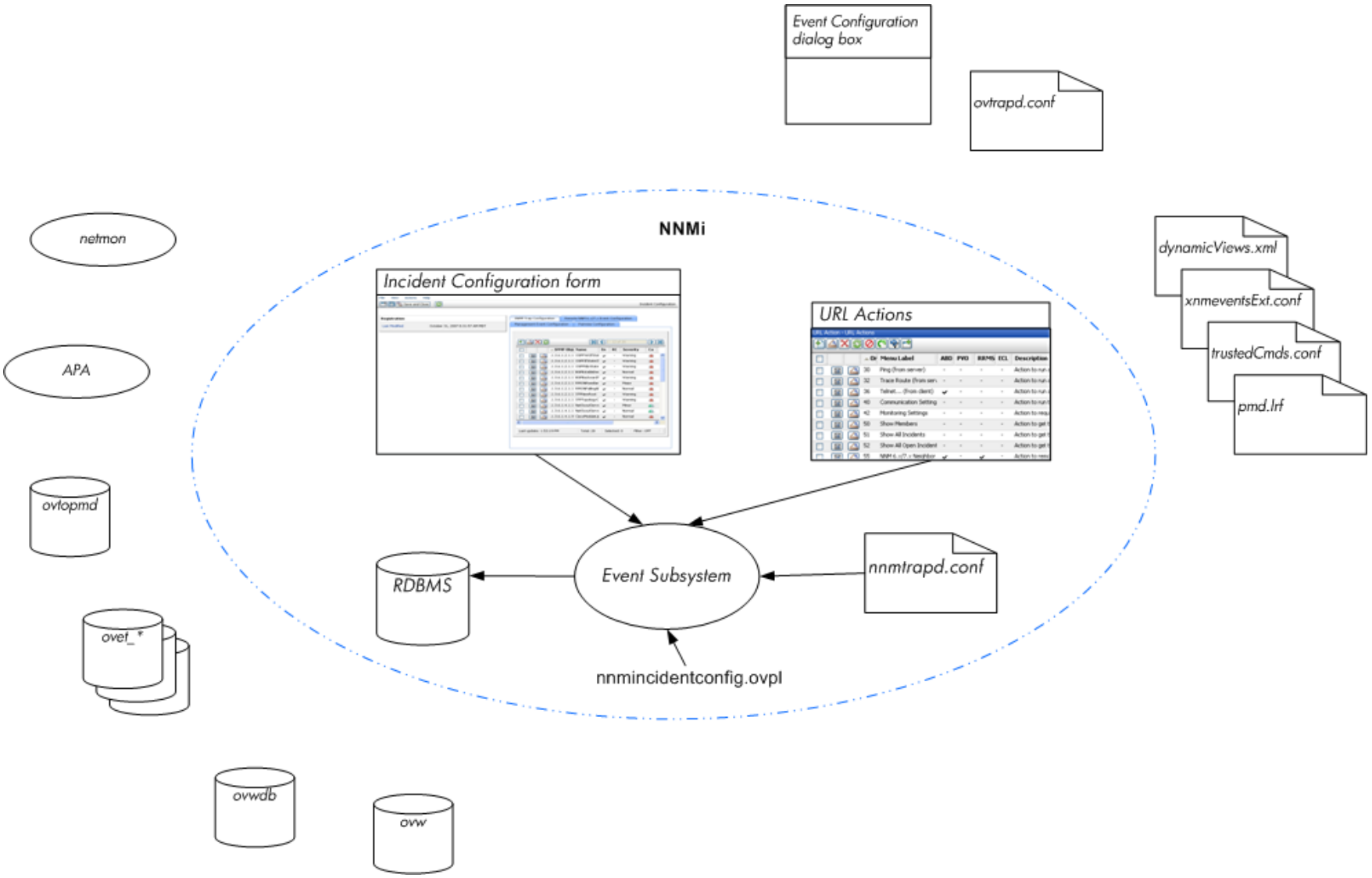
Customizing Event Monitoring

NNMi provides one centralized location, the incident views, where the management events, SNMP traps, and NNM 6.x/7.x forwarded events are visible to your team. You control which SNMP traps and NNM 6.x/7.x events are considered important enough to appear as incidents.

The center of Figure 16 shows the tools, files, and commands used to configure event monitoring in NNMi. The perimeter of the figure shows the similar items for NNM 6.x/7.x.

The APA information applies to NNMi 7.x Advanced Edition only.

Figure 16 Event Monitoring Configuration Elements

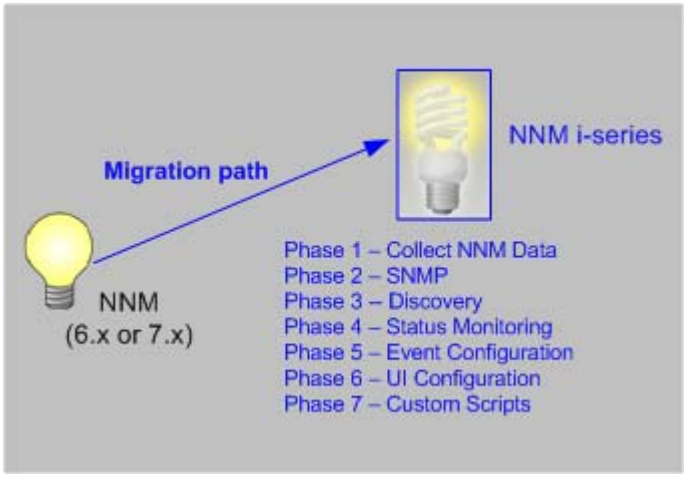


Key Concepts for Event Monitoring

This section briefly describes the main areas of change from NNM 6.x/7.x to NNMi. For more information about NNMi incidents, refer to *Configuring Incidents* in the NNMi help.

- In NNMi, the event subsystem is not used for inter-process communication and the volume of events is significantly reduced. The administrator no longer needs to configure whether each IPC message should be displayed or logged.
- NNMi receives only those traps for which it is configured. Unconfigured traps are filtered out of the event pipeline.
- NNMi displays all traps that it receives.
- Trap filters for the NNMi event subsystem processes are configured implicitly based on the selections in the **Incident Configuration** form.
- The NNMi `nnmincidentconfig.ovpl` command loads only the trap definitions from the named MIB files.
- NNMi provides pairwise, rate, and de-duplication correlations that occur in the event pipeline. (NNMi does not include the event correlation system (ECS).)
- In NNMi, you can configure actions that occur at any point in the lifecycle of an incident. Actions can be any script, executable, or Jython action.
- The NNMi URL Actions configuration replaces the `dynamicViews.xml` and `xnmeventsExt.conf` configuration files for defining actions that can be taken as a result of an event.

Upgrading from NNM 6.x/7.x



This chapter provides a basic migration path from HP Network Node Manager 6.x or 7.x to the most recent version of the HP Network Node Manager i Software, as indicated in the footer of this document. This path should serve the needs of most users. This chapter does not cover advanced migration topics or customizations; consulting services are available to meet your needs in these areas.

This chapter uses the following product naming conventions:

- **NNM** refers to older versions of HP Network Node Manager (including all 6.x and 7.x releases of NNM).
- **NNMi** refers to HP Network Node Manager i Software (including all 8.x releases of NNMi and NNMi Advanced).

This chapter makes the following assumptions:

- You have installed NNMi following the instructions in the *NNMi Installation Guide*.
- You have reviewed the concepts described in the NNMi help and the deployment information in this guide for a general understanding of NNMi functions.
- You understand how to use the NNMi console.

The information in this chapter will be updated frequently as migration tools are released and as NNMi evolves.

For up-to-date, downloadable copies of NNM and NNMi documentation, go to:

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

This chapter contains the following topics:

- Phase 7: Upgrade Custom Scripts
- Upgrade Tools Reference

Upgrade Options

A Fresh Beginning

Many NNM installations have been in place for several generations of software and in a variety of networking environments. Users who began with NNM 4.x or 5.x, in a routed world, might be carrying forward excess baggage that really does not apply to the current network structure. If your NNM installation is more than 2 years old, seriously consider using this opportunity to begin with a fresh installation. Completely re-evaluating how to manage your current network might result in a significant overhead drop and a streamlined operation compared with your NNM environment.

If you choose to start with a fresh installation of NNMi, install NNMi by following the instructions in the *NNMi Installation Guide*. Then consider the more complex deployment tasks presented in other chapters of this *NNMi Deployment Guide*. You do not need to read this migration chapter.

Upgrading in Phases

For some organizations, a phased approach to migration works better than a new installation. These organizations require that the new NNMi implementation completely reproduce and replace the existing NNM implementation. While there are many possible paths to that end, HP recommends the following phases:

- Phase 1: Collect Data from the NNM Management Station

  Use the NNMi-provided tools to gather the information needed for migration from the NNM management station.

- Phase 2: Upgrade SNMP Information

  Configure NNMi with the SNMP access information for your environment.

- Phase 3: Upgrade Discovery

  Configure NNMi to discover the objects that were discovered by NNM by approximating the way that NNM discovered them (automatically).

- Phase 4: Upgrade Status Monitoring

  Configure the status polling intervals and protocols that are most appropriate for your environment.

- Phase 5: Upgrade Event Configuration and Event Reduction

  Configure NNMi to display the event severity, category, message, and to perform the automatic actions you had configured in NNM. You might also need to configure deduplication, rate counting, pairwise cancellation, and threshold monitoring.

- Phase 6: Upgrade Graphical Visualization

  Select one of the following approaches:

  — Phase 6: Upgrade Graphical Visualization (OVW)

  Configure NNMi with node group maps that are similar to the NNM OVW location submaps.

  — Phase 6: Upgrade Graphical Visualization (Home Base)

  Configure NNMi with node group maps that are similar to the NNM 7.x Advanced Edition Home Base container views.

- Phase 7: Upgrade Custom Scripts

  Update scripts that use NNM command line tools to call NNMi command line tools.

NNMi can act as a manager of managers for your existing NNM systems. You can configure NNM to forward events to NNMi. Then you can use the NNMi console, with its consolidated user interface, incident ownership, and lifecycle states to navigate to familiar NNM tools. For instructions on integrating NNM into NNMi, see Integrating NNM 6.x or NNM 7.x with NNMi 8.11 on page 243.

Table 21 presents a high-level overview of the migration process for the two ends of the migration complexity continuum:

- The simplest approach involves importing environment-specific information from NNM and accepting the default NNMi configuration values, which are improved from NNM.
- The most detailed and thorough approach takes a close look at the NNM configuration and replicates this configuration in NNMi.

The remainder of this chapter walks through the process of migrating an NNM configuration to NNMi. The text in the left margin indicates how the specific steps fit into the migration process:

- **Gather from NNM** indicates work to be done on the NNM management station.
- **Replicate to NNMi** indicates work to be done on the NNMi management server.
- **Enhance in NNMi** indicates optional work to be done on the NNMi management server. You can perform enhancements during the migration process or at any time in the future.

At appropriate points, you will be given two or more options along the complexity continuum for completing a given task.

Table 21 Upgrade Continuum

Phase	Simplest Approach	Most Detailed and Thorough Approach
Collect Data from NNM	1 Use the NNMi-provided tools on the NNM management station. 2 Copy the collected data to the NNMi management server.	1 At each migration phase, gather the appropriate NNM configuration data by hand. 2 Copy the collected data to the NNMi management server.
SNMP Information	Import the collected community strings into NNMi, and let NNMi sort out which community string goes with which node.	1 Export all community strings currently in use. 2 Modify the data file and import the contents to NNMi as specific node community strings.
Discovery	Modify the collected list of discovered nodes, and import the file contents into NNMi as seeds with no auto-discovery rules.	1 Determine how NNM and `netmon` find nodes (seeds, loadhosts, filters, other tools). 2 Replicate this approach as closely as possible with seeds and auto-discovery rules.
Status Monitoring	NNMi defaults are updated to match most customer requirements. You might not need to make significant changes to these default values, so begin with the updated default values.	1 Determine exactly what polling intervals and polling policies were used by NNM and `netmon` or APA for each group of nodes. 2 Implement NNMi node groups and interface groups to replicate the polling intervals and polling policies.
Event Configuration and Event Reduction	1 Start with the default configuration from NNM. 2 Add the definitions for any custom traps from managed devices. 3 Add automatic actions as necessary.	1 Determine exactly what NNM customizations have been made for each trap and event type. 2 Customize each matching trap and event type on the NNMi system.
Graphical Visualization	1 Import the NNM `ovw` containers. 2 Assign node groups to containers. **OR** 1 Import the NNM 7.x Advanced Edition container views. 2 Assign node groups to containers.	1 In the most inclusive NNM map, determine what is on each submap. 2 Create a node group for the contents of each NNM submap. 3 For each node group, create an NNMi map, add a background image, and place each node.
Custom Scripts	Modify existing scripts to use the `nnmtopodump.ovpl` command.	Write new scripts that incorporate the new tools in NNMi.

Phase 1: Collect Data from the NNM Management Station

As of NNMi 8.10 patch 1, NNMi provides tools that run on the NNM management station to collect the majority of data needed for migrating the NNM configuration to NNMi. The tools create text files from information in the NNM databases and copy other configuration information. The tools also assemble the data into a known directory structure for copying to the NNMi management server.

NNMi 8.1x patch 3 adds the `nnmetmapmigration.ovpl` tool and revises the `nnmtopodump.ovpl` command.

For information about the data collection tools and the information that these tools collect, see Data Collection Tools on page 239.

Gather from NNM

Upgrade tool approach

1. Perform a complete back up of the NNM system.
2. Copy the data collection tool archive from the NNMi management server to the NNM management station. The file name and locations depend on the operating system of each computer.
   - On the NNMi management server, the archive is in the following directory:
     - *Windows*: `%NnmInstallDir%\migration\`
     - *UNIX*: `$NnmInstallDir/migration/`
   - On the NNM management station, place the archive as follows:
     - *Windows*: Copy the `migration.zip` file to the NNM installation folder (*`install_dir`*, usually similar to `C:\Program Files\HP OpenView`).
     - *UNIX*: Copy the `migration.tar` file to the `/opt/OV/` directory.
3. Unpack the data collection tool archive using a tool or command that is appropriate for the operating system of the NNM management station.
4. From the NNM installation directory, run the tools:
   a. Change to the `migration` directory.
   b. Create the expected directory structure for the data to be collected:

      `createMigrationDirs.ovpl`

   c. Collect the NNM data:

      `nnmmigration.ovpl`

   d. If you want to include the OVW map location hierarchy data in the upgrade archive, complete the upgrade tool approach for gathering the map data as described in Phase 6: Upgrade Graphical Visualization (OVW) on page 236.

   If Home Base container views are configured on the NNM management station, this information is included in the upgrade archive. No additional work is necessary.

e Archive the collected data:

archiveMigration.ovpl

This tool creates the *<hostname>*.tar file of the collected data for simple data transfer to the NNMi management server. The tool consumes a large amount of memory while it is running. If the NNM system does not have enough available memory or disk space, this tool fails; you can archive the data yourself in smaller chunks or copy individual files as needed.

On Windows operating systems, archiveMigration.ovpl may run slowly. Consider using another tool for archiving the data in preparation for moving it to the NNMi system.

Manual approach

If the upgrade tool approach does not work in your environment, follow the steps listed in each phase for gathering NNM data at that time.

Replicate to NNMi

Copy the data archive to the NNMi management server.

Upgrade tool approach

If the archiveMigration.ovpl tool completed successfully, follow these steps:

1 On the NNMi management server, change to the following directory:

- *Windows*: %NnmDataDir%\tmp\
- *UNIX*: $NnmDataDir/tmp/

2 In the tmp directory, create the migration and *<hostname>* directories in the following structure:

- *Windows*: %NnmDataDir%\tmp\migration*<hostname>*\
- *UNIX*: $NnmDataDir/tmp/migration/*<hostname>*/

3 Copy the *<hostname>*.tar file from the NNM management station to the following location on the NNMi management server:

- *Windows*: %NnmDataDir%\tmp\migration*<hostname>**<hostname>*.tar
- *UNIX*: $NnmDataDir/tmp/migration/*<hostname>*/*<hostname>*.tar

4 On the NNMi management server, change to the directory that you created in step 2:

- *Windows*: %NnmDataDir%\tmp\migration*<hostname>*\
- *UNIX*: $NnmDataDir/tmp/migration/*<hostname>*/

5 Unpack the data archive:

- *Windows*:

  **%NnmInstallDir%\migration\bin\restoreMigration.ovpl **
  -source *<hostname>*.tar

- *UNIX*:

  **$NnmInstallDir/migration/bin/restoreMigration.ovpl **
  -source *<hostname>*.tar

Manual approach

If the `archiveMigration.ovpl` command did not complete successfully, copy the data files manually.

The process of copying a text file from Windows to UNIX can insert `^M` characters into the file.

- To avoid this problem, transfer files using FTP in ASCII mode.
- To remove `^M` characters from a text file, on the UNIX system run the `dos2ux` (or similar) command.

Phase 2: Upgrade SNMP Information

Migrate the SNMP community string information that NNMi uses to establish connections with managed devices.

If the NNM configuration includes IP addresses or hostnames that should not be looked up in the name resolution service, migrate that information to NNMi.

Customize NNMi device profiles for the custom devices in your network.

Configure SNMP Access

NNMi discovery requires SNMP access to the managed nodes in order to collect specific information about their configuration and connectivity. SNMP is also used during status monitoring to assess the health of the node and the objects it contains.

NNM tries community strings serially, in the order listed for the matched region, and uses the first one that works. NNMi tries all configured community strings in parallel and uses the first one that works. Use the best community string where there might be multiple working values.

Gather from NNM

Upgrade tool approach

The `nnmmigration.ovpl` tool collected the community strings from the NNM management station into the `snmpCapture.out` file.

Manual approach

The NNM management station has the complete configuration information for SNMP access to the equipment in your environment.

1 Export the NNM SNMP configuration by doing one of the following:

- Open a user interface, select **Options** → **SNMP Configuration**, and then click **Export**. Name the target file `snmpout.txt`.
- Run the command:

```
xnmsnmpconf -export > snmpout.txt
```

NNM SNMP information example

Your output will look something like the following example:

```
10.2.126.75:public:*::::::
mytest57.example.net:public:*::::::
127.0.0.1:public:*::::::
10.97.233.209:mycommstr:*::::::
mpls2950.example.net:mycommstr:*::::::
mplsce04.example.net:mycommstr:*::::::
*.*.*.*:mycommstr:*:8:2:900:::
```

The target file contains the following fields separated by colons:

```
target:community:proxy(* indicates do not proxy):timeout
(tenths of a second):retries:poll interval
(seconds):port:set-community:
```

To see a clear interpretation of the values (but not for use in importing), use the command:

```
xnmsnmpconf -export -verbose
```

For a description of the `ovsnmp.conf` file format, see the *ovsnmp.conf* reference page, or the UNIX manpage, on the NNM management station.

2 Review any configured alternative community strings in the following file:

- *Windows*: `%OV_CONF%\netmon.cmstr`
- *UNIX*: `$OV_CONF/netmon.cmstr`

Replicate to NNMi

Upgrade tool approach

1 Change to the following directory:

- *Windows*: `%NnmDataDir%\tmp\migration\<hostname>\SNMP\`
- *UNIX*: `$NnmDataDir/tmp/migration/<hostname>/SNMP/`

2 Create a text file of the NNM community strings:

- *Windows*:

```
%NnmInstallDir%\migration\bin\snmpCapture.ovpl \
snmpCapture.out > snmpout.txt
```

- *UNIX*:

```
$NnmInstallDir/migration/bin/snmpCapture.ovpl \
snmpCapture.out > snmpout.txt
```

3 Follow one of the manual approaches for loading the community strings into NNMi.

4 Configure timeout, retries, and port in the NNMi console.

Manual approaches

Choose an approach to entering community strings into NNMi. Each of these approaches starts with the list of unique community string values in the `snmpout.txt` file that you created in step 2 on page 208 (for the upgrade tool approach) or step 1 on page 207 (for the manual approach).

The `SNMP proxy system` and `Set community name` configuration areas are not transferable.

Simple manual approach

The easiest approach is to enter all NNM community strings and let NNMi determine the SNMP community string to use for each device. Community string discovery is enabled by default; you can use this feature to expedite migration.

1 Notify your network operations center (NOC) to expect authentication errors during NNMi's initial discovery. NOC personnel can safely ignore these authentication errors during that time.

2 Complete one of the following actions:

- Modify the `snmpout.txt` file to match the format used by NNMi. Then use NNMi to load these values.
- Use the `snmpout.txt` file as a sample and hand-build the input file for NNMi. Then use NNMi to load these values.
- Enter the values in the NNMi console by following these steps:

a Determine the list of unique community string values in the `snmpout.txt` file.

If you used the upgrade tool approach to create the `snmpout.txt` file from the `snmpCapture.out` file, each community string in the `snmpout.txt` file is unique; you do not need to perform this step.

— *Windows*: Open the `snmpout.txt` file in Microsoft Office Excel. Select the data rows, and then sort on column B.

For this example, consider two unique community strings:

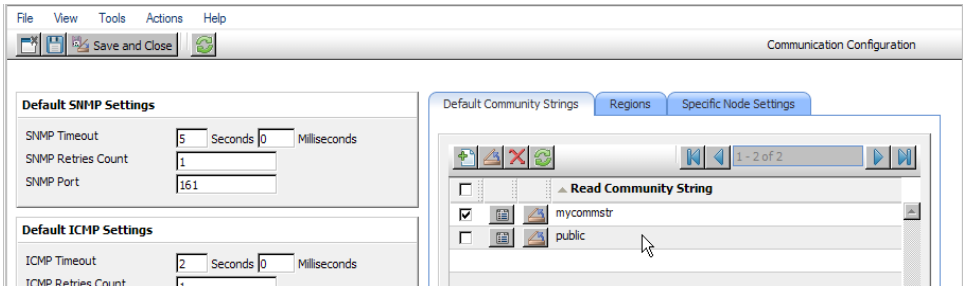
```
public
mycommstr
```

— *UNIX*: Run the following command:

```
cut -f 2 -d ':' < snmpout.txt | sort -u
```

b In the NNMi console, select **Communication Configuration** from the **Configuration** workspace. Enter all of the unique values on the **Default Community Strings** tab.

c Configure timeout, retries, and port.



Modified simple manual approach

Group community strings by IP region where they are used. Load regional values into the NNMi console, and then let NNMi determine the SNMP community string to use for each device, but with fewer authentication failures than in the simple approach.

1 In the `snmpout.txt` file, determine the list of unique values *per IP region* that NNM is using.

2 In the NNMi console, select **Communication Configuration** from the **Configuration** workspace. Create IP Regions, and then enter the community strings for each region.

3 Configure timeout, retries, and port.

Automated manual approach

Convert the `snmpout.txt` file into the format needed by the `nnmcommload.ovpl` command, and then load the specific community string in use for each device.

1 Adapt the `snmpout.txt` file for use with the NNMi tool by using one of the following methods:

- Use an editor to create the file appropriate for NNMi. The result should look similar to:

```
10.2.126.75,public
mytest57.example.net,public
127.0.0.1,public
10.97.233.209,mycommstr
mpls2950.example.net,mycommstr
mplsce04.example.net,mycommstr
```

- *UNIX only*: Run the following command:

```
awk 'BEGIN {FS = ":" };{printf"%s,%s\n",$1,$2 }' \
<snmpout.txt> mysnmp.txt
```

  This command works for individual nodes in the file. Trim ranges or wildcards out by hand.

2 Run the following command:

```
nnmcommload.ovpl -u username -p password -file mysnmp.txt
```

3 Configure default community strings and community strings for IP ranges in the NNMi console.

4 Configure timeout, retries, and port in the NNMi console.

NNMi console approach

In the NNMi console, select **Communication Configuration** from the **Configuration** workspace. Duplicate the configured values from the `snmpout.txt` file.

Enhance in NNMi

Enhance your communication access configuration in NNMi with the following information:

- Hostname wildcards (if they suit your environment better than IP ranges)
- ICMP timeout and retries by global default, IP range, and specific node
- Enable or disable SNMP or ICMP access to specific areas of the network
- The preferred management address for specific nodes

When NNM selects a management address, it selects the lowest loopback address. NNMi 8.1x also selects the lowest loopback address. NNMi 8.0x selects the highest loopback address.

Limit Name Resolution

If you know of limitations in your DNS (or other name resolution) service, you can instruct NNM and NNMi to avoid lookups for those devices. If this task does not apply to your installation, continue to Customize Device Profiles on page 212.

Gather from NNM

Upgrade tool approach

The `nnmmigration.ovpl` tool collected the information about which IP addresses and hostnames to use without DNS lookup from the NNM management station and created one or both of the `ipNoLookup.conf` and `hostNoLookup.conf` files for configuring NNMi.

Manual approach

1. Review the following file to determine the **addresses** that NNM excludes from address-to-hostname resolution:
   - *Windows*: `%OV_CONF%\ipNoLookup.conf`
   - *UNIX*: `$OV_CONF/ipNoLookup.conf`

   If the `ipNoLookup.conf` file does not exist on the NNM management station, there is no configuration to replicate.

2. Run the following command to determine the **hostnames** that NNM excludes from name-to-address resolution:

   ```
   snmpnolookupconf -dumpCache > snmpnolookup.out
   ```

   If the `snmpnolookup.out` file is empty, there is no configuration to replicate.

Replicate to NNMi

Upgrade tool approach

1. If available, edit the `ipNoLookup.conf` and `hostNoLookup.conf` files created by the `nnmmigration.ovpl` tool to delete any references to the NNMi management server:
   - *Windows*:
     - `%NnmDataDir%\tmp\migration\<hostname>\CONFIG\ipNoLookup.conf`
     - `%NnmDataDir%\tmp\migration\<hostname>\DNS\hostNoLookup.conf`
   - *UNIX*:
     - `$NnmDataDir/tmp/migration/<hostname>/CONFIG/ipNoLookup.conf`
     - `$NnmDataDir/tmp/migration/<hostname>/DNS/hostNoLookup.conf`
2. Place the edited configuration files into the following directory:
   - *Windows*: `%NnmDataDir%\conf\`
   - *UNIX*: `$NnmDataDir/shared/nnm/conf/`

Manual approach

1 Add the addresses from the NNM `ipNoLookup.conf` to the following file:

- *Windows*: `%NnmDataDir%\conf\ipNoLookup.conf`
- *UNIX*: `$NnmDataDir/shared/nnm/conf/ipNoLookup.conf`

Do not add the IP address of the NNMi management server.

2 Add the hostnames that NNM excludes (from the `snmpnolookup.out` file that you created in step 2 on page 210) to the following file:

- *Windows*: `%NnmDataDir%\conf\hostNoLookup.conf`
- *UNIX*: `$NnmDataDir/shared/nnm/conf/hostNoLookup.conf`

Do not add the hostname of the NNMi management server.

For information on the format of these configuration files, refer to the *ipNoLookup.conf* and *hostNoLookup.conf* reference pages, or the UNIX manpages.

Enhance in NNMi

NNMi does lookups during discovery only. By replicating the NNM no-lookup configuration to NNMi, the spiral discovery operation is automatically enhanced.

In NNMi, you can choose to use the DNS hostname, IP Address, or MIB II `sysName` as the displayed name label. To do so, follow these steps:

1 In the NNMi console, open the **Configuration** workspace.

2 Select **Discovery Configuration.**

3 Set your node name preferences in the **Node Name Resolution** area.

Customize Device Profiles

NNM collects some configuration information directly from SNMP queries to the device. Other information is *derived* from the device's **system object ID** (`sysObjectID`). Mapping from the `sysObjectID` to attributes in NNMi is done through **device profiles**. Device profiles are used to group nodes for monitoring, to filter nodes for viewing, and to categorize nodes for discovery maintenance.

The following configuration areas are not transferable:

- Custom symbols
- Custom database fields and default values

Gather from NNM

1 Determine any customizations to the OID files for your version of NNM.

- NNM 6.4 and earlier used the files `oid_to_sym`, `oid_to_type`, and `HPoid2type` to map a system's `sysObjectID` to database attributes and displayed symbol.
- NNM 7.x replaces the `oid_to_sym` file with the `oid_to_sym_reg` directory structure.

The `nnmmigration.ovpl` tool copies these files to the `CONFIG` folder within the migration data structure.

Replicate to NNMi

Because NNMi ships with a large number of device profiles that are preconfigured for known system object IDs, the device profiles that you need might already be available. The simplest approach is to start the discovery process, review the results, and then make modifications only as necessary.

Best practice

HP recommends that you specify a unique author for each device profile that you create or modify in case you need to identify these profiles at a later time.

2 In the NNMi console, select **Device Profiles** from the **Configuration** workspace. Locate the entry by `sysObjectID` for each of your customized values.

3 Update the device profile configuration as necessary.

- For the entries that NNMi has available, verify that the configured values match the NNM attributes.
- For entries that are not included in NNMi, create a new device profile for the `sysObjectID`. Submit an enhancement request to notify HP to add the ID for future releases.

Best practice

4 After initial discovery, sort the node inventory by device profile to locate the **No Device Profile** nodes.

The **No Device Profile** profile type indicates `sysObjectID`s that were not previously configured in NNMi. NNMi uses the default monitoring settings for nodes with **No Device Profile**, and these nodes are more difficult to filter.

You can build new device profiles to ensure that configured device profiles exist for all `sysObjectID`s in the NNMi database.

Phase 3: Upgrade Discovery

Migrate the discovery schedule and configuration. NNMi spiral discovery begins immediately after you save one or more discovery seeds.

Configure NNMi to use the appropriate community strings for your network environment before initiating discovery.

After initial discovery, migrate any connections between devices that were configured manually in NNM.

Schedule Discovery

The NNM discovery processes can run independently. To migrate discovery to NNMi, you only transfer the **interval** at which NNM discovers nodes.

The following schedule configuration areas are no longer used in NNMi and are not transferable:

- Topology checks on connector devices. A topology check now happens automatically whenever NNMi sees a trigger that indicates a possible change.
- Configuration check. A configuration check now happens at the time of a scheduled discovery or with any trigger in NNMi.

- Layer 2 (Extended Topology) discovery behavior. NNMi performs Layer 2 discovery for each device as it is found, so there is no need to schedule this behavior separately.
- Auto-adjusting discovery polling interval.

Gather from NNM

1 Determine when NNM performs rediscovery.

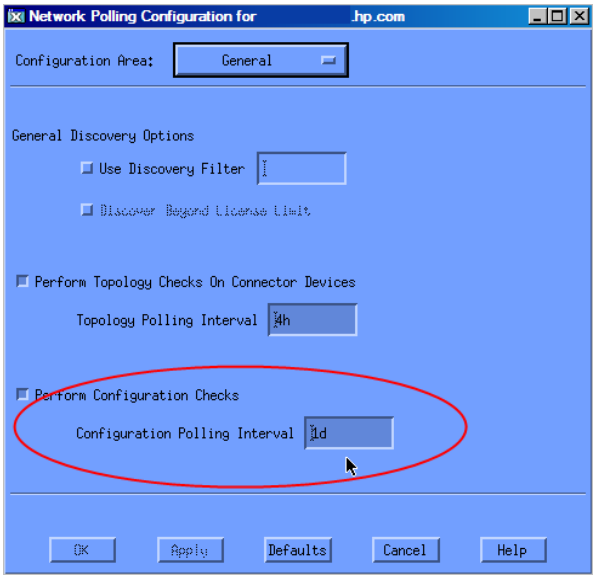
a In a user interface, select **Options** → **Network Polling Configuration**.

b On the **IP Polling** page, review the **Discovery polling interval** box.

— If NNM uses a fixed interval, note that value for transfer to NNMi.

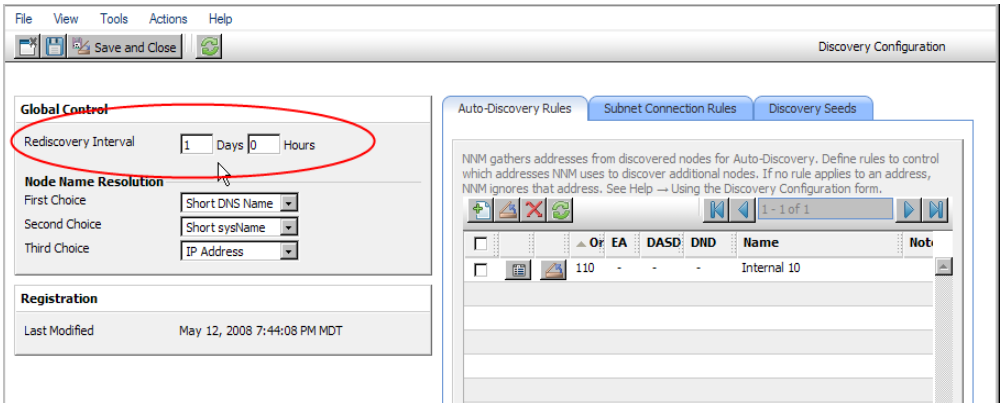
— If NNM uses auto-adjusting intervals, NNM waits a maximum of 24 hours. You can choose to stay with 24 hours, or you can select a new value.

— If auto-discovery has not been not enabled, determine the interval for **Perform configuration checks** on the **General** page and note that value for transfer to NNMi.



Replicate to NNMi

2 In the NNMi console, select **Discovery Configuration** from the **Configuration** workspace, and then set the **Rediscovery Interval** to the value determined in step 1.



Enhance in NNMi

All other configuration updates are automatic and incremental, so configuration is simpler and discovery is more efficient than in NNM.

Select Your Discovery Method

Determine which model to use for NNMi discovery:

- Seeded discovery with no auto-discovery rules. This type of discovery is bounded by the administrator, who controls what is discovered by adding seeds as necessary. Complete *only* the following task:
  - — Add Seeds to NNMi for Seeded Discovery on page 220
- Automatic discovery based on seeds and auto-discovery rules. Complete both of the following tasks:
  - — Configure Auto-Discovery Rules on page 215
  - — Add Seeds to NNMi for Seeded Discovery on page 220

For more information on the differences between the NNMi discovery methods, refer to *Determine Your Approach to Discovery* in the NNMi help.

NNM licenses are based on the number of nodes under management (status monitoring). NNMi licenses are based on the number of nodes discovered and placed in the topology (monitored and unmonitored nodes).

While this difference might encourage you to discover fewer nodes, there are advantages to including unmonitored nodes in your database. For example:

- You might want to see a service provider's access router and your connectivity to it, even if you are not responsible for managing the device.
- Status monitoring algorithms are based on connectivity as seen in the database. Interfaces having no device on the other end of the link *in the database* are unmonitored by default. You might choose to override the default in status monitoring configuration, or you might choose to discover the device. Your choice depends on the balance of interests in your environment. For more information, see Interfaces to Unmonitored Nodes on page 64.

Configure Auto-Discovery Rules

NNMi discovery configuration provides an excellent opportunity to consider what you want to manage with NNMi. Before you invest in converting your NNM discovery configuration and filters, consider looking at your current network environment and describing what you want to include in the NNMi topology.

If you do want to invest in direct conversion, NNMi discovery rules encompass two task sets from NNM: extending the scope of discovery and limiting the objects discovered within that scope.

For NNMi configuration, it is important to define all of the rules to extend and/or limit discovery before entering the seeds, which initiates the discovery process.

The following schedule configuration areas are no longer used in NNMi and are not transferable:

- IPX discovery from Windows
- Discover beyond license limit
- Disable discovery of Layer 2 objects (always enabled for NNMi)
- `netmon.interfaceNoDiscover`

- Discovery exclusions by filtering on attributes other than IP address and `sysObjectID` (and its derivatives)
- Limiting Layer 2 discovery through `bridge.noDiscover`
- Limiting Layer 2 discovery based on CDP protocol area (aggregated ports, vlans, etc.)
- Extended Topology zone configuration, which is no longer relevant to NNMi's spiral discovery

Configure Spiral Discovery

NNMi provides two methods for configuring spiral discovery in NNMi: manually loading nodes (for example, from a host file) and using auto-discovery rules.

Load nodes manually

Gather from NNM

1 In NNM, find the file that contains the output of the `loadhosts` command. This file lists an IP address and a hostname for each node, plus a subnet mask if one was specified.

NNM loadhosts example

An example file for the `loadhosts` command looks similar to the following:

```
10.2.32.201 lnt04.example.net  # comment
10.2.32.202 lnt07.example.net  # comment
10.2.32.203 lnt03.example.net  # comment
10.2.32.204 lnt02.example.net
10.2.32.205 lnt05.example.net
```

Replicate to NNMi

2 In NNMi, you can use discovery seeds in the same fashion as the NNM `loadhosts` command. To do so, use the `nnmloadseeds.ovpl` command with the `-f` option and specify a seed file.

Best practice

Complete all community string configuration prior to configuring any seeds into NNMi.

If you want the discovery output to be equivalent to NNM `loadhosts`, disable any auto-discovery rules that are configured in NNMi. To disable an auto-discovery rule, do one of the following:

- Delete the rule from the **Discovery Configuration** form.
- On the **Auto-Discovery Rule** form, clear the **Discover Included Nodes** check box.

The format for the seed file in NNMi is either an IP address or a node name (plus an optional comment) per line. For more information, refer to the *nnmloadseeds.ovpl* reference page, or the UNIX manpage.

NNMi seed file example

The following example shows an NNMi seed file with the same function as the NNM `loadhosts` command and a hostfile:

```
10.2.32.201 # comment
10.2.32.202 # comment
lnt03.example.net  # comment
lnt02.example.net
10.2.32.205
```

Best practice

The following file contains a list of devices from Extended Topology:

- *Windows*: `%OV_DB%\nnmet\hosts.nnm`
- *UNIX*: `$OV_DB/nnmet/hosts.nnm`

You can copy the first field (IP address) or second field (nodename) to create a seedfile for NNMi.

On UNIX, you can run the following command to create a file of the node names:

```
cut -f 2 hosts.nnm
```

Best practice

NNMi always favors the loopback address as the management address. If you do not use loopback addresses, NNMi probably (but not always) uses the seed address as the management address. Therefore, it is a good practice to populate the hostfile with preferred IP addresses. If you use hostnames, verify that the DNS resolves to the preferred management address, which still does not guarantee that NNMi will use this address as the management address. For more information about management address selection, refer to *Discovery Node Name Choices* in the NNMi help.

Use auto-discovery rules

Gather from NNM

1. Determine whether a discovery filter was used for NNM. In NNM, one discovery filter applied to the entire scope of discovery.
   a. Open an NNM user interface.
   b. Select **Options** → **Network Polling Configuration**.
   c. On the **General** page, review the **Use filter** check box and, if selected, note the discovery filter in use. If no filter is in use, continue with Add Seeds to NNMi for Seeded Discovery on page 220.
   d. Locate the discovery filter in the following file:
      — *Windows*: `%OV_CONF%\C\filters`
      — *UNIX*: `$OV_CONF/C/filters`
   e. Review the discovery filter logic carefully.

      For NNMi, you can filter on IP address ranges and system object ID ranges. You might be able to translate some attributes, such as hostname wildcards to IP ranges or vendor names to system object ID ranges.

NNM discovery filter example

The following example shows an NNM filter, including Routers, Bridges, Nokia_Firewalls, NetBotz, and NetsNSegs. You can see that NetBotz and Nokia firewalls are defined through their `sysObjectID`.

```
Nokia_Firewalls "Nokia Firewalls"
{ ( isNode && ( "SNMP sysObjectID" ~ .1.3.6.1.4.1.94.1.21.2.1.1 ) )
||
( isNode && ( "SNMP sysObjectID" ~ .1.3.6.1.4.1.94.1.21.2.1.9 ) )
||
( isNode && ( "SNMP sysObjectID" ~ .1.3.6.1.4.1.94.1.21.2.1.10 ) )
||
( isNode && ( "SNMP sysObjectID" ~ .1.3.6.1.4.1.94.1.21.2.1.11 ) )
||
( isNode && ( "SNMP sysObjectID" ~ .1.3.6.1.4.1.94.1.21.2.1.12 ) )
||
( isNode && ( "SNMP sysObjectID" ~ .1.3.6.1.4.1.94.1.21.2.1.138 ) )
}
```

```
NetBotz "NetBotz"
{ isNode && ( "SNMP sysObjectID" ~ .1.3.6.1.4.1.5528.* ) }

My_NetInfrastructure "My Network Infrastructure"
{ Routers || Bridges || Nokia_Firewalls || NetBotz || NetsNSegs }
```

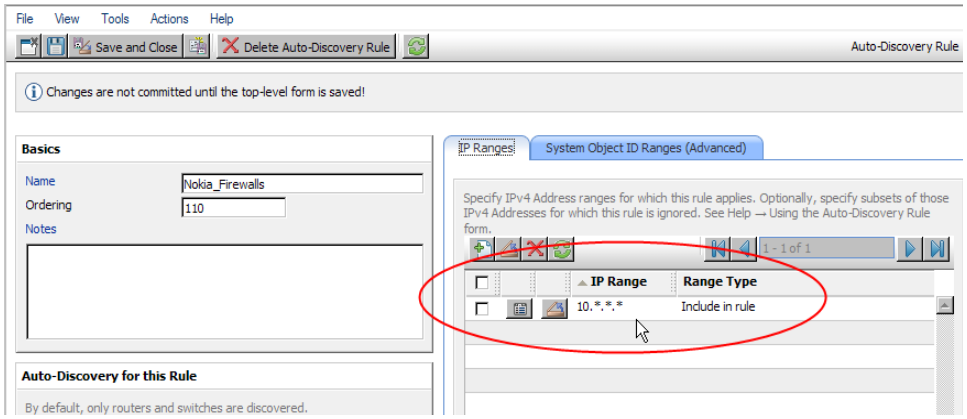
Replicate in NNMi

2 Enter the discovery filters in the NNMi console.

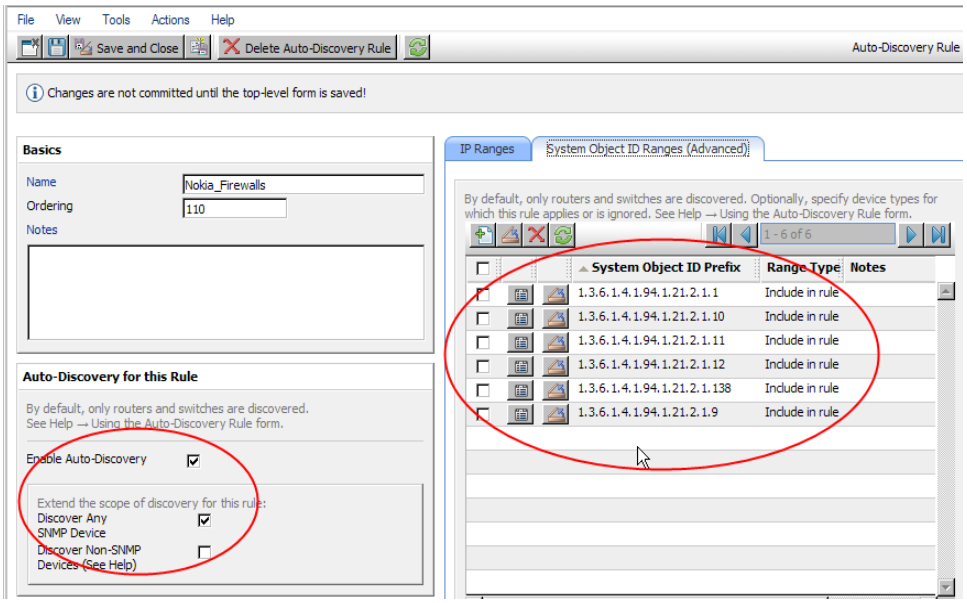
NNMi discovery filter entry example

For example, to transfer the NNM filter shown in the NNM discovery filter example on page 217 to NNMi, you would define three auto-discovery rules: one rule for Nokia firewalls, one rule for NetBotz devices, and a final rule for Routers and Switches (same as Bridge in NNM 7.x). NNMi does not require NetsNSegs. For this example, assume that the range of the network to be discovered is 10.*.*.*.

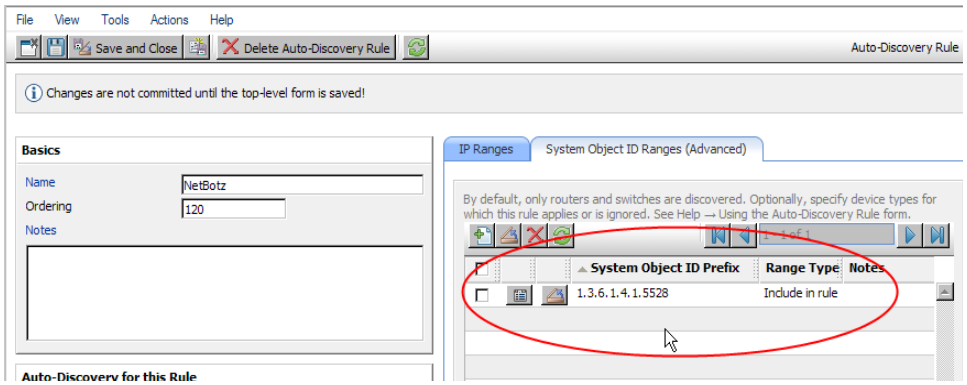
a For Nokia firewalls, enter a rule name (Nokia_Firewalls), and then enter the network IP range 10.*.*.*.



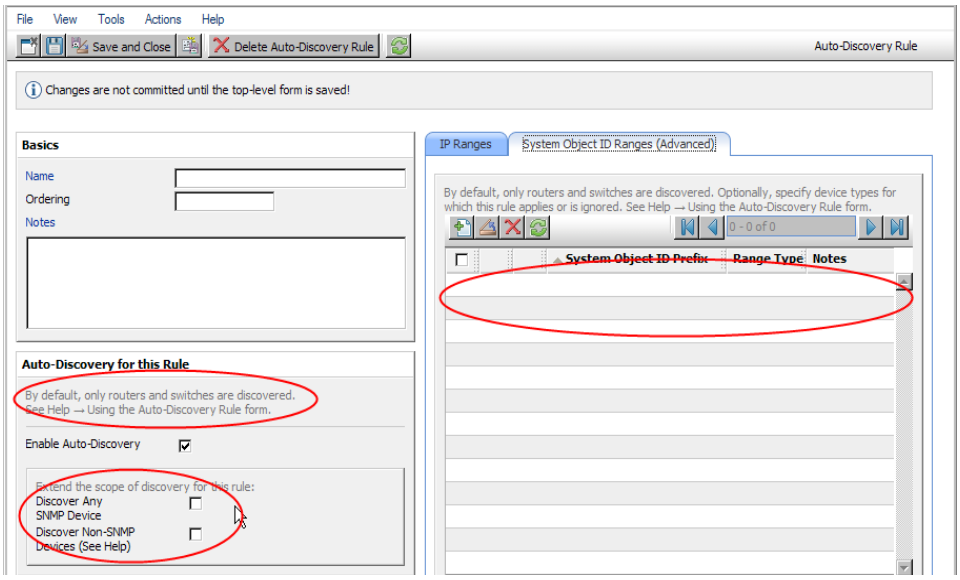
b Enter each `sysObjectID` (do not enter the leading period), and then select the **Discover Any SNMP Device** check box. (By default, NNMi only discovers switches and routers. Because these devices might not be marked as switches or routers, select the **Discover Any SNMP Device** check box when specifying `sysObjectIDs`.)



c Enter the NetBotz rule. This rule uses a wildcard in NNM: `.1.3.6.1.4.1.5528.*`. In NNMi the asterisk (.*) is implied and not required.



d The final rule is for switches and routers. Because NNMi discovers these devices by default, do not specify Object IDs (OIDs). You only need to specify the IP Range.



Exclude Addresses from Discovery

You can specify IP addresses that are never discovered. Do not populate the Excluded IP Addresses filter with the addresses associated with SNMPv1/SNMPv2c agents or SNMPv3 engines (the management addresses).

If the `netmon.noDiscover` file does not exist on the NNM management station, there is no configuration to replicate. You can follow the NNMi console approach to specify IP addresses that NNMi should not discover.

Gather from NNM

Upgrade tool approach

The `nnmmigration.ovpl` tool collected the `netmon.noDiscover` file from the NNM management station.

Manual approach

Review the following file to determine the IP addresses that NNM excludes from discovery:

- *Windows*: `%OV_CONF%\netmon.noDiscover`
- *UNIX*: `$OV_CONF/netmon.noDiscover`

Replicate to NNMi

Upgrade tool approach

1. Change to the following directory:
   - *Windows*: `%NnmDataDir%\tmp\migration\<hostname>\CONFIG\conf\`
   - *UNIX*: `$NnmDataDir/tmp/migration/<hostname>/CONFIG/conf`
2. Import the IP addresses in the `netmon.noDiscover` file into the NNMi database:
   - *Windows*:

     ```
     %NnmInstallDir%\bin\nnmdiscocfg.ovpl -excludeIpAddrs \
     -f netmon.noDiscover
     ```

   - *UNIX*:

     ```
     $NnmInstallDir/bin/nnmdiscocfg.ovpl -excludeIpAddrs \
     -f netmon.noDiscover
     ```

NNMi console approach

In the NNMi console, select **Discovery Configuration** from the **Configuration** workspace. On the **Excluded IP Addresses** tab, enter the IP addresses from the `netmon.noDiscover` file.

Add Seeds to NNMi for Seeded Discovery

Gather from NNM

Upgrade tool approach

The `nnmmigration.ovpl` tool collected the list of devices in the NNM database from the NNM management station into the `topology.out` file.

Manual approach

Determine the exact list of devices in the NNM database by running the following command:

```
ovtopodump > topology.out
```

Replicate in NNMi

1. Locate the `topology.out` (export) file from NNM.
   - For the upgrade tool approach, this file is located as follows:
     - *Windows*: `%NnmDataDir%\tmp\migration\<hostname>\TOPO\topology.out`
     - *UNIX*: `$NnmDataDir/tmp/migration/<hostname>/TOPO/topology.out`
   - For the manual approach, this file is in the local directory.

2 Copy and edit the `topology.out` file from NNM, or retype the entries into a file for importing into NNMi. The new file should have one explicit IP address or hostname per line. You do not need to specify a subnet prefix because NNMi determines the subnet automatically.

NNMi seed file example

```
10.2.32.201 # comment
10.2.32.202 # comment
lnt03.example.net  # comment
lnt02.example.net
10.2.32.205
```

Alternatively, you can add this list of nodes by using the NNMi console.

3 Run the following command:

`nnmloadseeds.ovpl -f newSeedfile`

For more information, refer to the *nnmloadseeds.ovpl* reference page, or the UNIX manpage.

NNMi begins to discover the devices associated with these seeds immediately and implements the existing device profiles (and node groups, such as node groups for status monitoring). NNMi spiral discovery is ongoing. For information on how to determine discovery status, refer to *Check Discovery Progress* in the *NNMi Installation Guide*.

Customize Connectivity

In certain circumstances where device information is limited, NNM's Extended Topology might not accurately discover and model every connection in a network. As a result, you might see no connections where you know connections exist or connections indicated where you know none exist. The remedy for this situation is to create the correct connections manually. You can replicate the connection configuration in NNMi.

Gather from NNM

1 Review the following file to determine whether manual connections have been configured in NNM:

- *Windows*: `%OV_CONF%\nnmet\connectionEdits`
- *UNIX*: `$OV_CONF/nnmet/connectionEdits`

The use of these files is documented in the *Using Extended Topology* manual or the white papers directory.

NNM connection example

The following example shows how to create two connections in NNM 7.x. One connection is based on `ifAlias`, and the other is based on `ifIndex` (along with board).

```
N1.example.net[ifAlias:MyAlias],N2.example.net[ifAlias:MyOtherAlias]
Y1.example.net[ 0 [ 999 ]],Y2.example.net[ 0 [ 2 ]]
```

Replicate to NNMi

2 Use the `nnmconnedit.ovpl` tool to make connection edits in NNMi. The file format is completely different from that used by NNM.

a Generate a connection template file by running the following command:

`nnmconnedit.ovpl -t addconn`

For more information, refer to the *nnmconnedit.ovpl* reference page, or the UNIX manpage.

b Edit the template file (`addconn.xml`) to change or add connections. Use the documentation in the file for the syntax of the new file.

NNMi connection example

The following example shows the NNMi equivalent to the NNM connection example on page 221:

```
<connectionedits>
        <connection>
                <operation>add</operation>
                <node>N1.example.net</node>
                <interface>MyAlias</interface>
                <node>N2.example.net</node>
                <interface>MyOtherAlias</interface>
        </connection>
        <connection>
                <operation>add</operation>
                <node>Y1.example.net</node>
                <interface>999</interface>
                <node>Y2.example.net</node>
                <interface>2</interface>
        </connection>
</connectionedits>
```

c Load the new connection information into the database by running the following command:

```
nnmconnedit.ovpl -f addconn.xml
```

d In the NNMi console, select **Layer 2 Connections** from the **Inventory** workspace to verify the results.

Phase 4: Upgrade Status Monitoring

In NNM 6.x, the `netmon` process performs status monitoring. In NNM 7.x, the `netmon` process or APA performs status monitoring.

- The `netmon` process models devices, such as nodes that contain interfaces, and applies polling parameters primarily at the node level.
- APA models addresses, interfaces, aggregated interfaces, boards, and nodes. APA can apply polling parameters at any of these levels.

With NNMi, you can apply polling parameters at the node, interface, or address level.

The following feature actions are no longer used in NNMi and are not transferable:

- Special handling for DHCP nodes
- Automatic deletion of a node that does not respond to polling
- Boards and aggregated ports, which are currently not modeled in NNMi and cannot have status monitoring

Set Polling Intervals

Gather from NNM

NNM netmon polling process

If the `netmon` process is your NNM general poller, obtain the polling intervals from the NNM user interface.

NNM APA polling process

NNM paConfig.xml example

If APA is your NNM general poller, find the `paConfig.xml` file and determine the current polling intervals. For example:

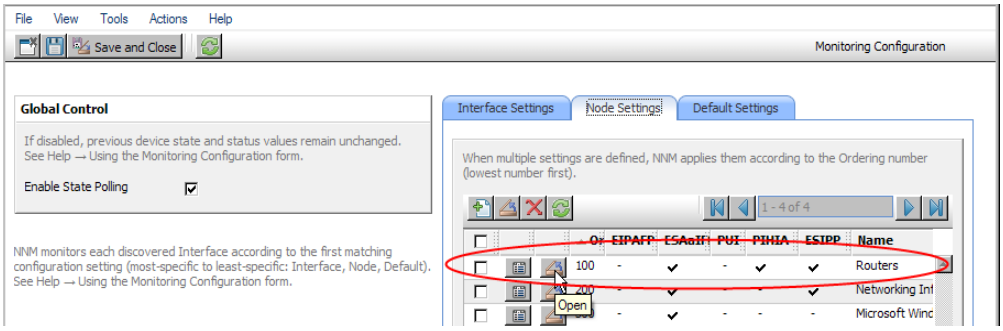
```
<classSpecification>
   <filterName>isRouter</filterName>
   <parameterList>
      <parameter>
         <name>interval</name>
         <title>Interval to Poll Device</title>
         <description>
            The interval for which the device will be polled
               in seconds.
         </description>
         <varValue>
             <varType>Integer</varType>
             <value>300</value>
         </varValue>
      </parameter>
          . . .
   </parameterList>
</classSpecification>
```

Replicate to NNMi

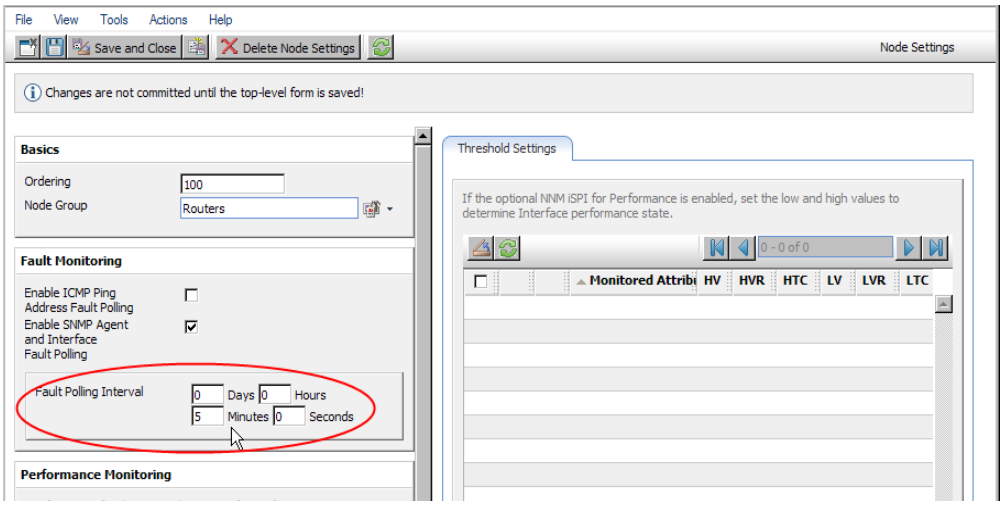
NNMi polling process

NNMi status monitoring configuration is based on groups of nodes and/or groups of interfaces.

1. In the NNMi console, select **Monitoring Configuration** from the **Configuration** workspace.
2. On the **Node Settings** tab, open a node group.



3 Set the **Fault Polling Interval** for the group.



Select Polling Protocol

Gather from NNM

NNM netmon polling process

By default, the `netmon` process uses ICMP to poll each address (equated with an interface). NNM can be configured so that the `netmon` process uses SNMP rather than ICMP (it never uses both) for some devices. To determine whether some areas are using ICMP, review the following file:

- *Windows*: `%OV_CONF%\netmon.snmpStatus`
- *UNIX*: `$OV_CONF/netmon.snmpStatus`

NNM APA polling process

APA uses a combination of SNMP and ICMP for polling. In APA, the polling policies are applied to nodes or interfaces, which are grouped by filters. The filters are defined in the `TopoFilters.xml` file. The polling policies are defined in the `paConfig.xml` file.

Replicate in NNMi

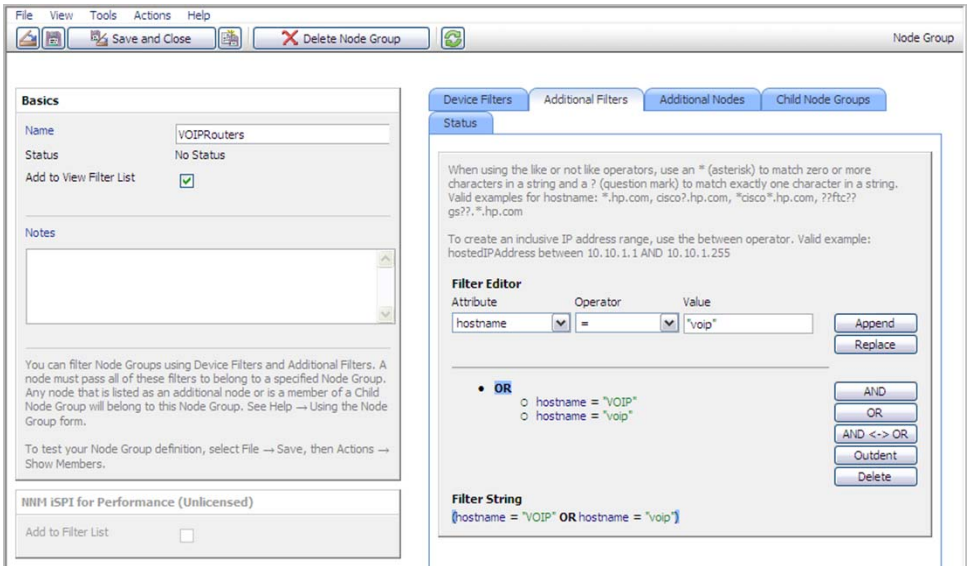
NNMi polling process

In NNMi, the nodes and interface collections are defined as node groups and interface groups. Polling policies are applied to node groups and interface groups on the **Monitoring Configuration** form.

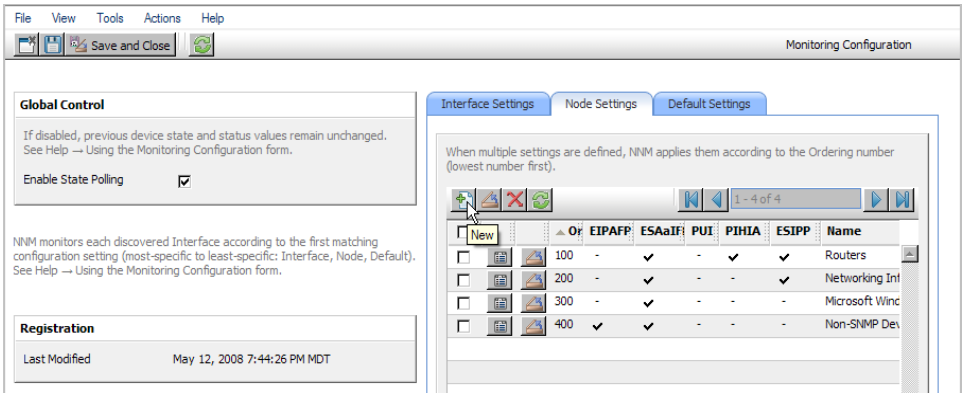
NNMi polling configuration example

For example, to configure polling (using SNMP and ping) for a collection of VOIP routers, follow these steps:

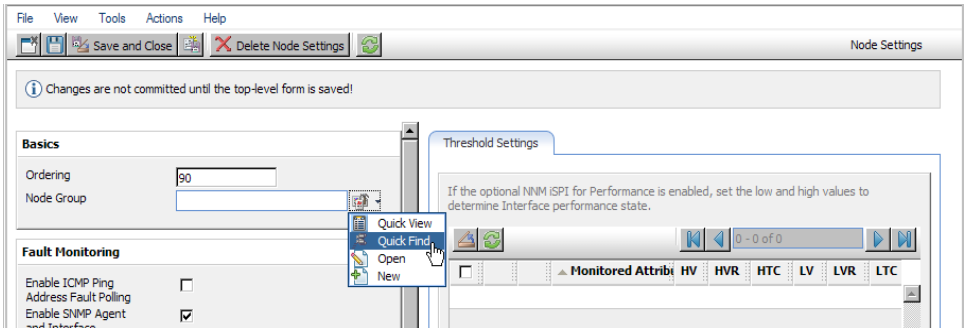
1 Using the **Node Group** form, create a node group that identifies the VOIP routers. Hostname wildcards are case sensitive, as shown here. Save and close this form.



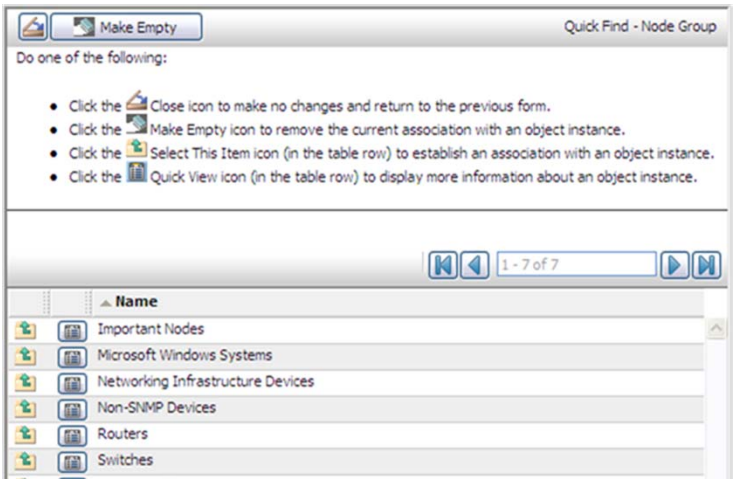
2 On the **Monitoring Configuration** form, add new node settings, as shown here.



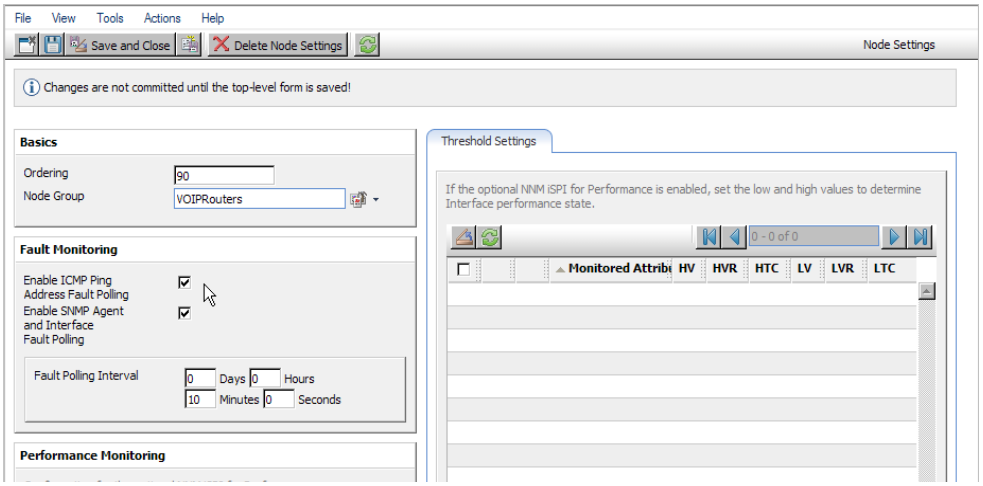
3 Specify an ordering value, and then select quick find for the **Node Group** field, as shown here.



4 Select the node group for these monitoring settings, as shown here.



5 Select the **Enable ICMP Ping Address Fault Polling** check box, as shown here. Save and close the form.



Configure Critical Nodes

By default, NNMi provides a node group for important nodes. This node group functions in the same way as the critical nodes list in NNM.

When important nodes are down or unreachable, NNMi shows node status as critical and generates a `NodeDown` incident.

Gather from NNM

NNM netmon polling process

If NNM uses `netmon` for status monitoring, NNM is not configured for critical nodes. You can create a new critical node configuration in NNMi.

NNM APA polling process

Review the following file to determine which nodes are designated as critical for APA:

- *Windows*: `%OV_CONF%\nnmet\topology\filter\CriticalNodes.xml`
- *UNIX*: `$OV_CONF/nnmet/topology/filter/CriticalNodes.xml`

NNM CriticalNodes.xml example

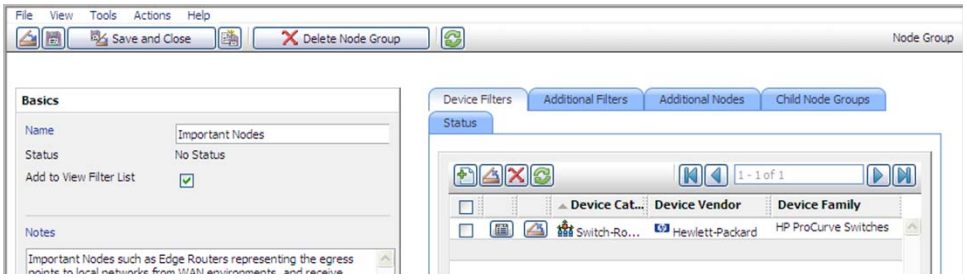
The `CriticalNodes.xml` file should resemble the following example:

```
<HostIDs xmlns="http://www.hp.com/openview/NetworkTopology/
TopologyFilter" xmlns:xsi="http://www.w3.org/2001/
XMLSchema-instance" xsi:schemaLocation="http://www.hp.com/openview/
NetworkTopology/TopologyFilter HostIDFile.xsd">
   <DNSName>router1.example.net</DNSName>
   <DNSName>router7.example.net</DNSName>
   <DNSName>MPLSRtr*.example.net</DNSName>
</HostIDs>
```

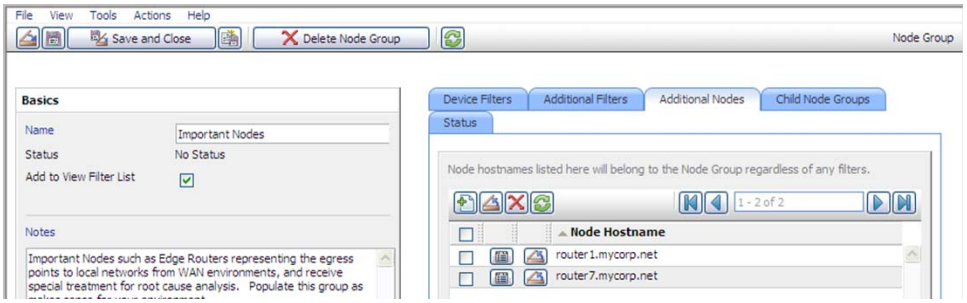
Replicate to NNMi

NNMi polling process

1. In the NNMi console, select **Node Groups** from the **Configuration** workspace.
2. Open the **Important Nodes** group.
3. Add the important nodes to the group by hostname wildcard, device filter, or specific nodes, as shown here.
   a. Add a device filter.



   b. Add specific nodes. Save and close the form.

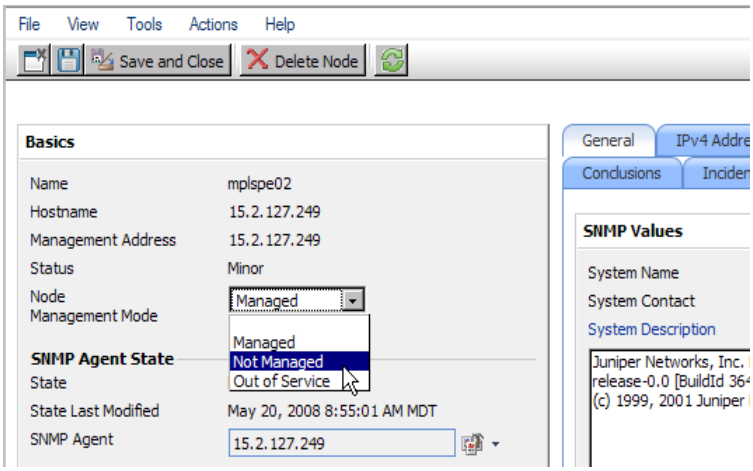


Exclude Objects from Status Polling

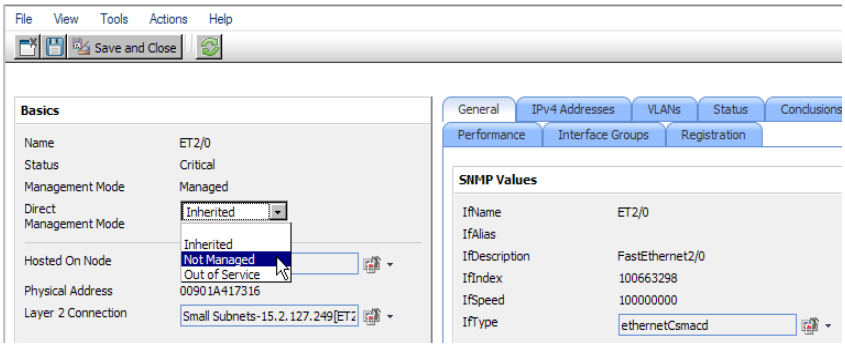
In NNM, most activities that stop nodes or interfaces from being monitored (set them to an UNMANAGED state) are completed through manual intervention in the NNM user interface.

NNMi streamlines the process of unmanaging objects. It is possible that the new product defaults match what you used to do manually (for example, only polling uplinks); however, managing settings through node groups and interface groups makes it easier to update settings automatically.

Occasionally you might need to mark a node or interface as **Not Managed**. You can set the management mode of an individual node on the **Node Details** form, as shown here:



You can set the management mode of an individual interface on the **Interface Details** form, as shown here:



Phase 5: Upgrade Event Configuration and Event Reduction

NNM analyzes all sources of incoming events (traps from managed devices, internal process communication, forwarded events) using an extended SNMPv2 format. Each event has an event object identifier, a name, and configuration parameters.

NNMi handles various sources of events differently. Traps from devices and events forwarded from NNM management stations are in the SNMPv2 format. Incidents created within NNMi contain names only; they do not have event object identifiers. In addition, NNMi internal process communications use a new (non-trap) mechanism to significantly improve overall performance. NNMi does not have `no format in trapd.conf` messages for unrecognized events. Unrecognized events are now discarded. If the NNM management station forwards events to the NNMi management server, ensure that NNMi contains incident definitions for all forwarded events.

Some Composer correlation types (suppress, enhance, transient, multisource) are no longer used in NNMi and are not transferable.

Display Traps from Devices

You can configure NNMi to display traps from devices in a way that is similar to the NNM environment.

NNMi contains default configurations for many of the common SNMP and vendor traps shipped with NNM. You can update NNMi with any customizations of these traps.

For a list of variables available for messages and automatic actions, refer to *Configure an Action for an Incident* and *Valid Parameters for Configuring Incident Actions* in the NNMi help.

Gather from NNM

Upgrade tool approach

The `nnmmigration.ovpl` tool collected the `trapd.conf` file and the MIBs that have been loaded into NNM.

Manual approach

Determine whether the NNM configuration includes customized traps. Note any customizations made to category, severity, display message, or automatic actions.

Replicate to NNMi

Upgrade tool approach

1. Change to the following directory:
   - *Windows*: `%NnmDataDir%\tmp\migration\<hostname>\CONFIG\conf\`
   - *UNIX*: `$NnmDataDir/tmp/migration/<hostname>/CONFIG/conf/`
2. Load the NNM MIBs into NNMi:
   - *Windows*:

     ```
     %NnmInstallDir%\migration\bin\nnmmibmigration.ovpl \
     -file snmpmib -u <user> -p <password>
     ```

   - *UNIX*:

     ```
     $NnmInstallDir/migration/bin/nnmmibmigration.ovpl \
     -file snmpmib -u <user> -p <password>
     ```

   This step only loads TRAP-TYPE and NOTIFICATION-TYPE MIB entries. NNMi does not use other MIB variables.
3. Load the NNM event definitions that are not included out-of-the-box for NNMi:
   - *Windows*:

     ```
     %NnmInstallDir%\migration\bin\nnmtrapdload.ovpl \
     -loadTrapd <lang>\trapd.conf -authorLabel NNM_migration \
     -authorKey com.domain.nnmUpgrade -u <user> -p <password>
     ```

   - *UNIX*:

     ```
     $NnmInstallDir/migration/bin/nnmtrapdload.ovpl \
     -loadTrapd <lang>/trapd.conf -authorLabel NNM_migration \
     -authorKey com.domain.nnmUpgrade -u <user> -p <password>
     ```

Best practice

HP recommends that you specify a unique author for this operation in case you need to identify these event definitions at a later time.

Manual approach

1 Download the vendor MIB files to the NNMi management server.

2 Run the following command for each MIB:

```
nnmincidentcfg.ovpl -loadTraps mibFile
```

- If one MIB has a dependency on another MIB file, use the following command to preload the dependencies:

  ```
  nnmincidentcfg.ovpl -loadMib mibFile
  ```

  Alternatively, you can use the **nnmloadmib** command, and then rerun `nnmicidentcfg.ovpl` with the `loadTraps` option.

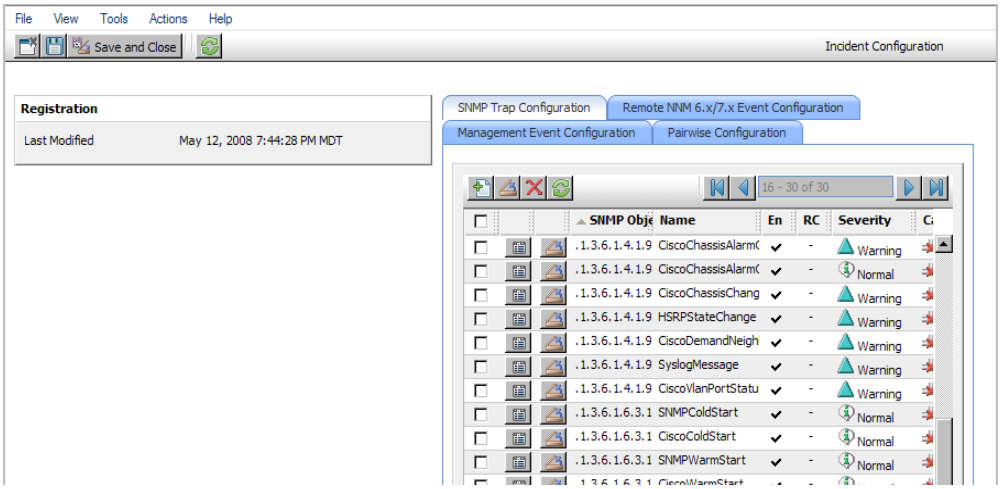
- To see which MIBs are already loaded, use the command:

  ```
  nnmloadmib -list
  ```

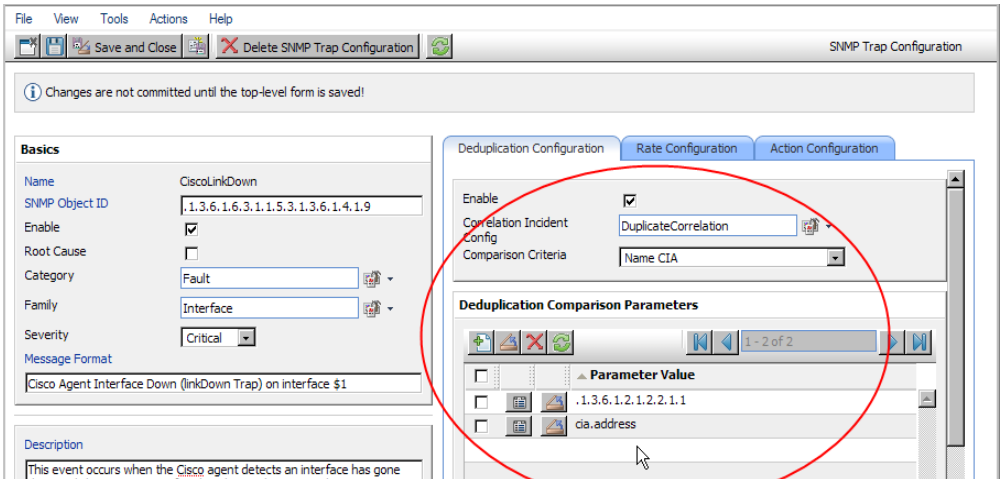
For more information, refer to the *nnmincidentcfg.ovpl* and *nnmloadmib* reference pages, or the UNIX manpages.

These steps only load TRAP-TYPE and NOTIFICATION-TYPE MIB entries. NNMi does not use other MIB variables.

3 In the NNMi console, select **Incident Configuration** from the **Configuration** workspace. Then select the **SNMP Trap Configuration** tab.



4 Customize the trap displays to match those in NNM. You can create categories as needed on the trap configuration form.



Enhance in NNMi

5 (Optional) In addition to setting default **Severity**, **Category**, and **Message**, set a default **Family**.

6 (Optional) Classify the trap as a root cause, so that it will appear in the **Root Cause Incidents** view.

Customize Display of NNMi-Generated Management Events

In NNMi, event configuration is simplified because the NNMi causal engine generates a more concise root cause than NNM.

You can modify the incidents generated with NNMi so that they have a similar appearance to NNM alarms. For example, you can customize the NNMi `NodeDown` incident message to be similar to the message for an NNM `NodeDown` alarm.

Gather from NNM

1 In NNM, determine any customizations to the events configuration.

Replicate to NNMi

2 In the NNMi console, select **Incident Configuration** from the **Configuration** workspace. Then select the **Management Event Configuration** tab.

3 Locate the new incident configuration by name rather than event number.

4 *Optional*. Customize event displays to match those in NNM by creating categories on the trap configuration form.

5 In addition to setting default **Severity**, **Category**, and **Message**, you can set a default **Family**.

Block/Ignore/Disable Traps

NNM provides several levels of event processing:

- Block traps as they come into `ovtrapd`
- Process, but do not store or display traps or events labeled `IGNORE`
- Store and process (correlate) events labeled `LOGONLY`, but never display them
- Store, process, and display an event into a category
- Traps that arrive without a configuration appear in the Alarm Browser as `No format in trapd.conf for…` and are stored in the database

NNMi has a simpler approach. A *disabled* event or trap is not stored, processed, or displayed. An *enabled* event or trap is fully stored, processed, and displayed. Any event for which NNMi does not have a configuration is blocked.

Gather from NNM

Upgrade tool approach

The `nnmmigration.ovpl` tool collected the `ovtrapd.conf` file.

The `ovtrapd.conf` file is available for NNM 7.51 or higher. The upgrade tool approach does not consider trap definitions. You might want to manually port the `LOGONLY` configuration for NNM traps.

Manual approach

1 Determine any customizations that ignore traps or set traps to `LOGONLY`.

2 Determine whether NNM uses the trap filtering mechanism (`ovtrapd.conf`, new with NNM 7.51).

Replicate to NNMi

Upgrade tool approach

1. Change to the following directory:
   - *Windows*: `%NnmDataDir%\tmp\migration\<hostname>\CONFIG\conf\`
   - *UNIX*: `$NnmDataDir/tmp/migration/<hostname>/CONFIG/conf/`
2. Copy the non-commented lines from the NNM `ovtrapd.conf` file into the `nnmtrapd.conf` file:
   - *Windows*:

     ```
     %NnmInstallDir%\migration\bin\nnmtrapdMerge.ovpl \
     ovtrapd.conf
     ```

   - *UNIX*:

     ```
     $NnmInstallDir/migration/bin/nnmtrapdMerge.ovpl \
     ovtrapd.conf
     ```

Manual approach

1. In the NNMi console, select **Incident Configuration** from the **Configuration** workspace. Locate any events that you do not want to receive or display, and clear the **Enable** check box for those events.
2. To block traps from specific IP addresses, edit the following file to update NNMi with the trap filtering information from NNM:
   - *Windows*: `%NnmDataDir%\shared\nnm\conf\nnmtrapd.conf`
   - *UNIX*: `$NnmDataDir/shared/nnm/conf/nnmtrapd.conf`
3. Use the `nnmtrapconfig.ovpl` command to enable trap blocking and to configure the rates and thresholds for trap blocking.

   For information about using this command, refer to the *nnmtrapconfig.ovpl* reference page, or the UNIX manpage.

Configure Automatic Actions

Gather from NNM

1. Determine any automatic actions that have been configured for NNM.

Replicate to NNMi

2. Copy action scripts from the NNM management station to the NNMi management server, where file location is not important.
3. In the NNMi console, select **Incident Configuration** from the **Configuration** workspace.
4. For each NNM event with an automatic action, configure the corresponding NNMi incident with that action (on the **Action Configuration** tab).

   To match the behavior of NNM, set the **Lifecycle State** to **Registered**.

Enhance in NNMi

5 Note the following NNMi configuration techniques:

- You can configure more than one automatic action to occur when an event arrives.
- You can configure one or more additional actions for each of the other lifecycle states (**In Progress**, **Completed**, **Closed**).
- You can pass more incident attributes to the command than in NNM.
- The procedure is simplified because you do not need to register commands in a separate configuration file before NNMi can run them.

Configure Additional (Manual) Actions

NNM provides operator actions or additional actions that are available from the menu in the Alarms Browser. You might be able to simulate the NNM actions with URL actions that are available from the NNMi console menu.

Gather from NNM

1 Determine any custom operator actions that are in NNM.

Replicate to NNMi

2 For these customer actions, determine how to transfer them to be available as URLs.

3 In the NNMi console, select **URL Actions** from the **Configuration** workspace.

4 Click **New**.

5 Provide the **Menu Label**, a **Unique Key**, **Ordering**, **Selection Type**, and the **Full URL** for the action.

Event Correlation: Repeating Events

NNM mechanisms use either the first or last event as the parent when deduplicating events.

NNMi creates a new parent and displays it when you select **Stream Correlated Incidents** from the **Incident Browsing** workspace. The original events appear in their configured view.

Gather from NNM

1 Determine whether the `RepeatedEvents` correlation is in use for NNM.

2 Determine whether the `Repeated` correlator is in use for NNM.

3 Determine whether deduplication is in use (`dedup.conf` file).

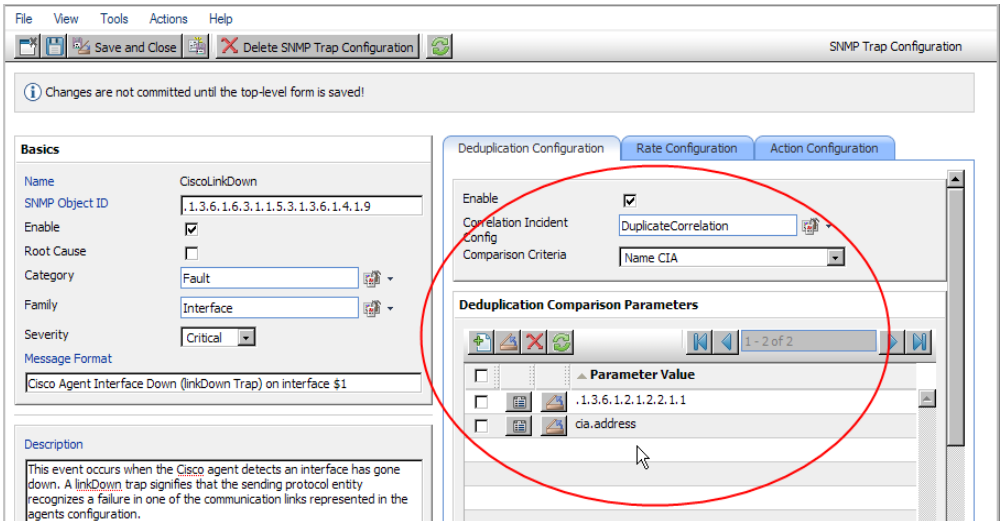
Replicate to NNMi

4 In the NNMi console, select **Incident Configuration** from the **Configuration** workspace.

5 Open the event to be deduplicated.

6 Select the **Deduplication Configuration** tab, enable deduplication, select a new parent event, and then define the matching criteria.

Deduplication in NNMi has no time limit.



Event Correlation: Counting the Rate

NNM mechanisms use either the first or last event as the parent when deduplicating events.

NNMi creates a new parent and displays it when you select **Stream Correlated Incidents** from the **Incident Browsing** workspace. The original events appear in their configured view, and NNMi has sustained rate behavior equivalent to the rolling time window in NNM.

Gather from NNM

1 Determine whether the rate correlator is in use for NNM.

Replicate to NNMi

2 In the NNMi console, select **Incident Configuration** from the **Configuration** workspace.

3 Select the **Management Event Configuration** tab.

4 Open the event identifier to be counted.

5 Select the **Rate Configuration** tab, and then do the following:

a Select **Enable** to enable monitoring.

b Set the count window.

c Set the time window (**Hours**, **Minutes**, and **Seconds** fields).

d Select a new parent event (**Correlation Incident Config**).

e Define the **Comparison Criteria**.

For more information, refer to *Management Event Form* in the NNMi help.

Event Correlation: Pairwise Cancellation

NNMi does not limit cancellation to a specific time window.

Gather from NNM

1. Determine whether the PairWise correlation is in use in NNM.
2. Determine whether the Transient correlator is in use in NNM.

Replicate to NNMi

3. In the NNMi console, select **Incident Configuration** from the **Configuration** workspace.
4. Select the **Pairwise Configuration** tab.
5. Select an existing pair, or click **New**.
6. Configure the paired event identifiers and the matching criteria.

For more information, refer to *Configuring Incidents* in the NNMi help.

Event Correlation: Scheduled Maintenance

NNMi can suppress the monitoring of unavailable nodes. To do so, use the OUT OF SERVICE mode. Unlike NNM, you cannot schedule OUT OF SERVICE maintenance in advance, and you must manually return the objects to MANAGED mode.

SNMP traps sent by devices in OUT OF SERVICE mode are suppressed in NNMi.

If your organization has been using the Scheduled Maintenance correlation, you can use the list of systems that are taken offline together.

Gather from NNM

1. Determine whether the `ScheduledMaintenance` correlation is in use in NNM.

Replicate to NNMi

2. In the NNMi console, select **Node Group Configuration** from the **Configuration** workspace.
3. Create a node group for each set of nodes in the **NNM Maintenance List**. Set the node groups to be available as view filters.
4. When it is time for maintenance, in the NNMi console select **Nodes** from the **Management Mode** workspace.
5. Filter the view to a specific node group by using the **Set node group filter** selector at the top.
6. Select all nodes, and then select **Actions > Out of Service**.
7. After maintenance is completed, select the nodes, and then select **Actions > Manage**.

Phase 6: Upgrade Graphical Visualization (OVW)

In NNM, an OVW map consists of multiple submaps, each of which shows a location or subnet in the network hierarchy. The NNM administrator can define multiple OVW maps and assign a different OVW map to each user.

In NNMi, topology maps are based on the defined node groups. While some topology maps may have a hierarchical relationship, such hierarchy is not limited to network subnets and locations. Additionally, all users can access all available topology maps.

The NNMi migration tools can replicate into NNMi the location submap hierarchy of one OVW map. Because the map structure is very different between the two products, the migration tools do not transfer nodes, networks, or leaf node elements from NNM.

Gather from NNM

Upgrade tool approach

1. Ensure that the migration tools have been set up as described in Phase 1: Collect Data from the NNM Management Station on page 205.
2. Set or create the `PERL5LIB` environment variable to the following value:
   - *Windows*: *`install_dir`*`\migration\lib`
   - *UNIX*: `/opt/OV/migration/lib`
3. Identify and open the NNM map that is most representative of the location hierarchy that you want to use in NNMi.
4. In the open map, click **File > Export** to create a map data file with the following name and location:
   - *Windows*: *`install_dir`*`\migration\ipmap.out`
   - *UNIX*: `/opt/OV/migration/ipmap.out`
5. Change to the following directory:
   - *Windows*: *`install_dir`*`\migration\`
   - *UNIX*: `/opt/OV/migration/`
6. Process the map data file:
   - *Windows*:

     ***`install_dir`*`\migration\bin\nnmmapmigration.ovpl ipmap.out`**
   - *UNIX*:

     `/opt/OV/migration/bin/nnmmapmigration.ovpl ipmap.out`

   This command creates the `nnmnodegrouplist.csv` and `backgrounds.tar` files, which are available in the following location:
   - *Windows*: *`install_dir`*`\migration\<hostname>\MAPS`
   - *UNIX*: `/opt/OV/migration/<hostname>/MAPS`

Replicate in NNMi

Upgrade tool approach

1. If you have not already done so, copy the `nnmnodegrouplist.csv` and `backgrounds.tar` files from the NNM management server to the following location:
   - *Windows*: `%NnmDataDir%\tmp\migration\<hostname>\MAPS\`
   - *UNIX*: `$NnmDataDir/tmp/migration/<hostname>/MAPS/`

2 Change to the following directory:

- *Windows*: `%NnmDataDir%\tmp\migration\<hostname>\MAPS\`
- *UNIX*: `$NnmDataDir/tmp/migration/<hostname>/MAPS/`

3 Import the node group definitions for the NNM location hierarchy into the NNMi database:

- *Windows*:

```
%NnmInstallDir%\bin\nnmloadnodegroups.ovpl -u <user> \
-p <password> -r false -f nnmnodegrouplist.csv
```

- *UNIX*:

```
$NnmInstallDir/bin/nnmloadnodegroups.ovpl -u <user> \
-p <password> -r false -f nnmnodegrouplist.csv
```

4 Make the NNM background graphics available to NNMi:

a Unpack the `backgrounds.tar` file using a tool or command (such as `restoreMigration.ovpl`) that is appropriate for the operating system of the NNMi management server.

b Copy the extracted files to the following location:

— *Windows*: `%NnmDataDir%\shared\nnm\www\htdocs\images\`

— *UNIX*: `$NnmDataDir/shared/nnm/www/htdocs/images/`

Alternatively, you can transfer the individual image files to the `images` directory using FTP in ASCII mode.

5 In the NNMi console, apply the appropriate background graphic to each location node group map:

a In the NNMi console, select **Node Groups** from the **Configuration** workspace.

b Examine the text in the **Notes** column.

If the migration tool created the node group, the note field indicates that it was created from an OVW location symbol. If the OVW submap included a background graphic, the note also specifies the image name.

c Select a migrated node group, and then click then **Actions > Node Group Map**.

d In the map, click **Save Layout**  to create a node group settings object for this node group.

e In the same map, click **File > Open Node Group Map Settings**.

f On the **Background Image** tab of the **Node Group Map Settings** form, specify the background graphics file that is identified in the note text in the **Node Groups** table for this node group, as described in step b.

On the **Node Group Map Settings** form, the path to the background graphics file is in the following format:

`/nnmbg/images/<optional_directory_structure>/<filename>`

In the file system, `/nnmbg/images/` maps to:

— *Windows*: `%NnmDataDir%\shared\nnm\www\htdocs\images\`

— *UNIX*: `$NnmDataDir/shared/nnm/www/htdocs/images/`

(The path in the note text applies to the NNM management station.)

6 In the NNMi console, add one or more node groups to the lowest-level topology map in the location hierarchy.

Phase 6: Upgrade Graphical Visualization (Home Base)

In NNM 7.x Advanced Edition, the Home Base can include container views that organize the network topology.

In NNMi, topology maps are based on the defined node groups. While some topology maps may have a hierarchical relationship, such hierarchy is not limited to network subnets and locations. Additionally, all users can access all available topology maps.

The NNMi upgrade tools can replicate into NNMi the Home Base container view hierarchy. Because the map structure is very different between the two products, the upgrade tools do not transfer nodes, networks, or leaf node elements from NNM.

Gather from NNM

Upgrade tool approach

The `nnmmigration.ovpl` tool collected the container view configuration file from the NNM management station.

Replicate in NNMi

Upgrade tool approach

1 Change to the following directory:

- *Windows*: `%NnmDataDir%\tmp\migration\<hostname>\NNMET\`
- *UNIX*: `$NnmDataDir/tmp/migration/<hostname>/NNMET/`

2 Parse the container view configuration file to create a comma-separated node group list:

- *Windows*:

```
%NnmInstallDir%\migration\bin\nnmetmapmigration.ovpl \
containers.xml nnmcontainerlist.csv.txt
```

- *UNIX*:

```
$NnmInstallDir/migration/bin/nnmetmapmigration.ovpl \
containers.xml nnmcontainerlist.csv
```

3 Import the node group definitions for the NNM 7.x Advanced Edition Home Base container hierarchy into the NNMi database:

- *Windows*:

```
%NnmInstallDir%\bin\nnmloadnodegroups.ovpl -u <user> \
-p <password> -r false -f nnmcontainerlist.csv.txt
```

- *UNIX*:

```
$NnmInstallDir/bin/nnmloadnodegroups.ovpl -u <user> \
-p <password> -r false -f nnmcontainerlist.csv
```

4 In the NNMi console, add one or more node groups to the lowest-level topology map in the location hierarchy.

Phase 7: Upgrade Custom Scripts

NNM provides several command-line tools for reading the contents of the NNM databases. These tools can be used from the command line. They can also be incorporated into scripts that were created for your network environment.

As of NNMi 8.10 patch 1, the `nnmtopodump.ovpl` command is located in the `bin` directory. This command is an enhanced version of what was previously provided as an unsupported tool in the `support` directory. The updated `nnmtopodump.ovpl` command can generate textual output in a format very similar to that of the NNM `ovtopodump` command. Additionally, you might be able to replace other NNM commands in custom scripts with the `nnmtopodump.ovpl` command.

As of NNMi 8.1x patch 3, the `-legacy short` option is available for the `nnmtopodump.ovpl` command.

Gather from NNM

Replicate in NNMi

1. Copy all custom scripts for reading the NNM databases to a working directory.
2. Copy the working directory to the NNMi management server.
3. Examine each script for calls to any of the following commands:
   - `ovtopodump`
   - `ovobjprint`
   - `ovet_topodump.ovpl`
   - `ovdwquery`
4. As appropriate, update each script to call the `nnmtopodump.ovpl` command in place of the commands named in the previous step.

   The `nnmtopodump.ovpl` command is not a direct replacement for any of the NNM commands. Compare the `nnmtopodump.ovpl` output with the expected output, and modify each script as needed.
5. Test and revise each updated script until it produces the desired results.

For more information, see the *nnmtopodump.ovpl* reference page, or the UNIX manpage.

Upgrade Tools Reference

This section describes the tools that NNMi provides to assist with migrating an NNM 6.x or 7.x configuration to NNMi. This information is current for the product and patch version indicated in the footer of this document.

Data Collection Tools

Run the data collection tools on the NNM 6.x/7.x management station to gather the NNM configuration information into one place. The procedures for using these tools are described earlier in this chapter.

The data collection tools are delivered in the NNMi patch as two archive files (`migration.zip` for Windows operating systems, `migration.tar` for UNIX operating systems). After patch installation, the archive files are available in the following location:

- *Windows*: `%NnmInstallDir%\migration\`
- *UNIX*: `$NnmInstallDir/migration/`

The data collection tools are limited by the availability of commands on the NNM management station. In some cases, these tools will not run to successful completion. If a wrapper script fails, you can run the tools individually. If a single tool fails, you can replicate the intent of the tool (as described here) to collect the data yourself.

Table 22 lists the tools that are included in the data collection tools archive files.

Table 22 Upgrade Data Collection Tools in the NNMi Patch

Tool	Description
createMigrationDirs.ovpl	Creates the directory structure to hold the migration data that will be collected from the NNM management station. For more information, see NNM Configuration Data Files on page 241.
nnmmigration.ovpl	Collects the NNM configuration data. This tool is a wrapper script that runs most of the other tools described in this table.
archiveMigration.ovpl	Packs the collected data into a tar archive file (`<hostname>.tar`) for easy transfer to the NNMi management server.
captureLocale.ovpl	Determines the locale of the NNM management server so that the tools collect the correct version of localized configuration files.
hostnolookup.ovpl	Runs `snmpnolookupconf -dumpCache` to create a text file (`hostNoLookup.conf` in the `DNS` directory) of the hostnames that NNM discovery ignores.
nnmtopodump.ovpl	Runs `ovtopodump -lr` to create a text file (`ovtopodump.out` in the `TOPO` directory) snapshot of the topology database. This tool is different from the tool of the same name that is installed into the `bin` directory on the NNMi management server.
ovmapdump.ovpl	Runs `ovmapdump -l` for each OVW map to create a text file (in the `MAPS` directory) snapshot of that map database.
ovmibmigration.ovpl	Verifies that all MIBs defined in the `NNM snmpmib` file have been loaded into NNM.
ovwdbDump.ovpl	Runs `ovobjprint` to create a text file (`ovobjprint.out` in the `OVWDB` directory) snapshot of the object database that future migration tools might use.

Table 22 Upgrade Data Collection Tools in the NNMi Patch (cont'd)

Tool	Description
snmpCapture.ovpl	Runs `xnmsnmpconf -dumpCache` to create a text file (`snmpCapture.out` in the `SNMP` directory) snapshot of the SNMP configuration database. This tool is different from the tool of the same name that is described in Table 24.
trapdConfNodes.ovpl	Parses the `trapd.conf` file to create node lists (`EVENTS\NODES\*`) that future migration tools might use.
nnmmapmigration.ovpl	Parses the export file for an OVW map to identify node groups of the locations in that map (`nnmnodegrouplist.csv` in the `MAPS` directory) and to collect the background image files that are used on location submaps (`backgrounds.tar` in the `MAPS` directory). Run this command separately from the `nnmmigration.ovpl` wrapper script.

NNM Configuration Data Files

The data collection tools store files in the following location:

- *Windows*: *`install_dir`*`\migration\<hostname>\`
- *UNIX*: `/opt/OV/migration/<hostname>/`

Where *<hostname>* is the hostname of the NNM management station. Table 23 lists the contents of the *<hostname>* directory.

Table 23 File Structure of the Collected NNM Configuration Data

Directory	Contents
CONFIG	A copy of the NNM CONF directory
DNS	hostNoLookup.conf
EVENTS	All trapd.conf files in the NNM configuration Node lists
MAPS	Application registration files Symbol registration files A flat file of each map database
NNMET	(NNM 7.x Advanced Edition) containers.xml
OVW.MAPS	Output of the nnmmapmigration.ovpl tool
OVWDB	A flat file of the object database Field registration files

Table 23 File Structure of the Collected NNM Configuration Data (cont'd)

Directory	Contents
SNMP	Community strings
TOPO	A flat file of the topology database
WWW	The NNM web interface files

Data Import Tools for Upgrading

Table 24 lists the tools that the NNMi patch provides for importing NNM 6.x/7.x data into the NNMi database. The migration process also uses standard NNMi tools. For information about the standard tools, see the appropriate reference pages, or the UNIX manpages.

Table 24 Data Import Tools in the NNMi Patch

Tool	Description
restoreMigration.ovpl	Unpacks the NNM configuration archive created by `archiveMigration.ovpl` on the NNM 6.x/7.x management station.
nnmetmapmigration.ovpl	Parses the NNM 7.x Advanced Edition Home Base container view definition file (`containers.xml`) to identify node groups of the locations in that view for NNMi.
nnmmibmigration.ovpl	Runs `nnmincidentcfg.ovpl` to import the MIBs in the NNM `snmpmib` file into the NNMi database. This tool does not re-load any MIBs that are already loaded in NNMi.
nnmtrapdload.ovpl	Loads trap definitions from the NNM `trapd.conf` file into the NNMi database. This tool loads only the first definition that it encounters for each trap. It does not re-load any trap definitions that are already loaded in NNMi.
nnmtrapdMerge.ovpl	Merges all non commented lines in the NNM `ovtrapd.conf` file into the NNMi `nnmtrapd.conf` file.
snmpCapture.ovpl	Outputs the contents of the `snmpCapture.out` file to STDOUT, one community string per line. This tool is different from the tool of the same name that is described in Table 22.

Integrating NNM 6.x or NNM 7.x with NNMi 8.11

You can integrate the following HP Network Node Manager (NNM) 6.x/7.x functionality with HP Network Node Manager i Software (NNMi) 8.10:

- You can forward events from NNM 6.x/7.x to the NNMi 8.11 management server in order to use the NNMi 8.11 incident views for managing incident life cycle.
- You can open some NNM 6.x/7.x views from the NNMi 8.11 management server.

This integration is useful for controlling the rate of migration to the new NNMi 8.11 product.

This integration is also useful for large managed environments with many NNM 6.x/7.x management stations. If you do not need the new functionality in NNMi 8.11 throughout the network, you can maintain a few NNM 6.x/7.x management stations while using NNMi 8.11 as your primary network management tool.

You can also use the information in this chapter to integrate a third-party product with NNMi 8.11. That product must be able to generate SNMP v1, v2C, or v3 traps and send them to the NNMi 8.11 management server.

This chapter contains the following topics:

Configure Event Forwarding

To set up event forwarding from the NNM 6.x/7.x management station to an NNMi 8.11 management server, complete the following procedures in order:

- Step 1: Configure NNM 6.x/7.x to Forward Events to the NNMi 8.11 Management Server
- Step 2: (Optional) Use Node Level Filtering to Further Reduce Events
- Step 3: Add the NNM 6.x/7.x Management Station to the NNMi 8.11 Topology
- Step 4: (Optional) Save the Management Station Configuration
- Step 5: Verify NNM 6.x/7.x Incident Configuration in the NNMi 8.11 Console

Step 1: Configure NNM 6.x/7.x to Forward Events to the NNMi 8.11 Management Server

On the NNM 6.x/7.x management station, configure each event that you want forwarded to the NNMi 8.11 management server. Most of these events will be under the OpenView Enterprise. Interesting events include:

- OV_Node_Down (OV_Node_Up, Ov_Node_Unknown, and so forth)
- OV_APA_NODE_DOWN (OV_APA_NODE_Intermittent, and so forth)
- OV_Station_Critical (OV_Station_Normal, and so forth)
- OV_Error (OV_Warning, OV_Inform) information about system health
- OV_Message (OV_Popup_Message, and so forth)

For the complete list of recommended NNM 6.x/7.x events to forward, see the events listed on the **Remote NNM 6.x/7.x Event Configuration** tab of the **Incident Configuration** form in the NNMi console.

Recommended and Supported Procedure: Use the Event Configuration Window

If you do not have an XServer, see Alternative Procedure: Manually Edit trapd.conf on page 246.

To configure an NNM 6.x/7.x event to forward to the NNMi 8.11 management server, follow these steps:

1. At the command prompt, enter:

   ```
   ovw
   ```

   Alternatively, run `xnmtrap` from the command line, and then continue with step 3.

2. Click **Options > Event Configuration**.
3. In the **Event Configuration** window, select the **Openview** enterprise in the top pane, and then double-click an event name in the bottom pane.

   To sort the events by name, click **View > Sort > Event Name**.

4 Specify the NNMi 8.11 management server to receive the forwarded events.

Best practice

If you have created a destination list file, enter the complete path to this file in the **Destination** field. For information about the destination list file format, see Optional: Destination List File on page 245.

- *Windows*: On the **Forwarding** tab in the **Modify Events** window, enter the host name of the NNMi 8.11 management server in the **Destination** field.

  Click **Add**, and then click **OK**.

- *UNIX*: In the **Destination** field at the bottom of the **Event Configurator** window, enter the host name of the NNMi 8.11 management server.

If you do not see the **Destination** field, select the **Forward Event** option in the center of the window.

Click **Add**, and then click **OK**.

5 Repeat step 3 and step 4 until you have configured all of the events that you want to forward to an NNMi 8.11 management server.

6 Click **File > Save**.

NNM 6.x/7.x saves the changes to the event configurations and automatically re-reads the new event configuration.

Optional: Destination List File

If you want to forward several events to the same group of NNMi 8.11 management servers, you can create a file that lists the forward destinations.

The recommended location for the destination list file is:

- Windows: `%OV_CONF%\nnm8EventForwardDestinations.txt`
- UNIX: `$OV_CONF/nnm8EventForwardDestinations.txt`

The destination list file is a text file with the following format:

- Each line is either one node name or a comment line.
- The first character of a comment line is the `#` character.

For example:

```
# List of destination NNMi 8.11 Management Servers to receive events.
# This list should be small enough that it does not overwhelm the NNMi 8.11 operators.
# In general, the events should be node-related, so that Neighbor Views launched remotely
from the NNMi 8.11 management server are meaningful.
#
system1.domain.com
system2.comain.com
system3.domain.com
```

For more information, see the `trapd.conf` manpage.

After creating or changing the destination list file, run the following command to re-read it:

```
xnmevents -event
```

Alternative Procedure: Manually Edit trapd.conf

If you do not have an XServer, you can manually edit the FORWARD field for each event in the following file:

- Windows: %OV_CONF%\C\trapd.conf
- UNIX: $OV_CONF/C/trapd.conf

Specify either a single NNMi 8.11 management server or a destination list file. For example:

```
EVENT OV_Message .1.3.6.1.4.1.11.2.17.1.0.58916872 "Application Alert Alarms" Normal
FORMAT $3
FORWARD NNM8Server.domain.com
```

The FORWARD field might also include a list of the remote managers. For example:

```
FORWARD %REMOTE_MANAGERS_LIST% /etc/opt/OV/share/conf/nnm8EventForwardDestinations.txt
```

After editing the trapd.conf file, run the following command to force NNM to re-read the event configuration:

```
xnmevents -event
```

Step 2: (Optional) Use Node Level Filtering to Further Reduce Events

In NNM 6.x/7.x, you can configure a node list for certain events. When a node list is present, an event coming into the NNM 6.x/7.x management station matches an event configuration only if the event source is in the node list. Thus, an event will be forwarded to the NNMi 8.11 management server only if the event source is in the node list. A typical use case for a node list is to forward only specific events from important nodes to the NNMi 8.11 management server.

For information on creating a node list in NNM 6.x/7.x, see the information about the sources_list in the ovtrapd.conf manpage.

Step 3: Add the NNM 6.x/7.x Management Station to the NNMi 8.11 Topology

Include the NNM 6.x/7.x management station in the NNMi 8.11 topology so that the NNMi 8.11 management server receives an incident if the NNM 6.x/7.x management station goes down.

If the NNM 6.x/7.x management station is not already in the NNMi 8.11 **Nodes** inventory view, add the management station to the discovery seeds, and then wait for it to be discovered.

For information on how to add a node to the discovery seeds, refer to *Discovering Your Network* in the NNMi help.

Step 4: (Optional) Save the Management Station Configuration

To save the new configuration, run the following command:

```
nnmconfigexport.ovpl -u <user> -p <password> -c station \
-f <filename>
```

You can later import the backup by running the following command:

```
nnmconfigimport.ovpl -u <user> -p <password> -f <filename>
```

For information on these commands, refer to their respective reference pages, or the UNIX manpages.

Step 5: Verify NNM 6.x/7.x Incident Configuration in the NNMi 8.11 Console

Verify that the events you forwarded from NNM 6.x/7.x are configured (as incidents) in NNMi 8.11.

NNMi 8.11 provides many out-of-the-box incident configurations. To view these out-of-the-box configurations, follow these steps:

1. From the workspace navigation panel, select the **Configuration** workspace.
2. Open the **Incident Configuration** form.
3. Click the **Remote NNM 6.x/7.x Event Configuration** tab.

   This tab displays the out-of-the-box incident configurations. If you open an incident, you should see that the type is **Remotely Generated**.

If one or more of the events that you configured for forwarding from the NNM 6.x/7.x management station is not listed on the **Remote NNM 6.x/7.x Event Configuration** tab, add a new incident configuration for each missing event. For information about how to configure a new incident, refer to *Configuring Incidents* in the NNMi help.

The incident categories in NNM 6.x/7.x are different from those in NNMi 8.11. For information about the relationship between the NNM 6.x/7.x alarm categories and the NNMi 8.11 incident categories, see Mapping Categories.

Mapping Categories

In NNM 6.x/7.x, the pre-configured alarm categories are as follows:

- Error Alarms
- Threshold Alarms
- Status Alarms
- Configuration Alarms
- Application Alert Alarms

In NNMi 8.11, the pre-configured incident categories are as follows:

- Accounting
- Application Status
- Configuration
- Fault
- Performance
- Security
- Status

Table 25 lists the mapping of NNM 6.x/7.x alarm categories to NNMi 8.11 incident categories that HP suggests:

Table 25 Suggested Category Mappings

NNM 6.x/7.x Alarm Category	NNMi 8.11 Incident Category
Error Alarms	Application Status
Threshold Alarms	Performance
Status Alarms	Status
Configuration Alarms	Configuration
Application Alert Alarms	Application Status

Configure Remote View Launching

To set up the NNMi 8.11 management server to display NNM 6.x/7.x views on the NNMi 8.11 management server, complete the following procedures in order:

- Step 1: Install Java Run-Time Environment (JRE)
- Step 2: Create an NNM 6.x/7.x Management Station Entity in NNMi 8.11
- Step 3: (Optional) Configure Additional NNM 6.x/7.x Views

Step 1: Install Java Run-Time Environment (JRE)

Although NNMi 8.11 does not have any requirements for a Java Run-Time Environment (JRE), NNM 6.x/7.x views require version 1.4.2.xx of the JRE. To download and install the required JRE, go to:

http://java.sun.com/j2se/1.4.2/download.html

NNM 7.53 does support newer JRE versions. Refer to the NNM 7.53 release notes for more information.

Step 2: Create an NNM 6.x/7.x Management Station Entity in NNMi 8.11

Configure the NNMi 8.11 management server to associate events received from the NNM 6.x/7.x management station to an entity in NNMi 8.11. This configuration enables the launching of NNM 6.x/7.x Dynamic Views from the NNMi 8.11 management server. For example, you can select a Node Down from My7xSystem that is displayed in NNMi 8.11, and then launch the URLs back to My7xSystem.

It is important to use the Primary Address that matches the address that is encoded in the event sent by the NNM 6.x/7.x management station. If you are not sure about this address, look at the **RemoteSenderAddress** in the custom incident attributes for an incident that was forwarded from the NNM 6.x/7.x management station.

To set up an NNM 6.x/7.x management station configuration, follow these steps:

1 In the NNMi 8.11 console, from the workspace navigation panel, select the **Configuration** workspace.

2 Open the **Management Stations** form.

3 Click  **New**.

4 Enter the following information:

- **Name**—An identifier for the NNM 6.x/7.x management station represented by this configuration.
- **NNM Version**—The NNM version (6.x or 7.x) of the management station that you are configuring.
- **Address**—An IP address for the NNM 6.x/7.x management station. This IP address much be reachable from the NNMi 8.11 management server. You can find the IP address in either of the following ways:
  - Run `ovaddr` at the command line on the NNM 6.x/7.x management station.
  - Determine the custom incident attribute (CIA) of an incident that has been forwarded from the NNM 6.x/7.x management station.

    This method works only if you have already completed the procedures that are described in Configure Event Forwarding on page 244 and if a configured event has been generated on the NNM 6.x/7.x management station and forwarded to the NNMi 8.11 management server.

- **ovas (OV Application Server) Port**—The port number of the OpenView Application Server (ovas) for the NNM 7.x management station that you are configuring. On NNM 7.x management stations, the port number is usually 7510.

  The ovas port also applies to NNM 6.x with the Extended Topology add-on.

- **Web Server Port**—The port number of the web server for the NNM 6.x/7.x management station that you are configuring:
  - For NNM 6.x management stations on a Windows operating system, this port number is usually 80.
  - For NNM 6.x management stations on a UNIX operating system, this port number is usually 3443.
  - For NNM 7.x management stations on all operating systems, this port number is usually 3443.
- **Description**—A description of the NNM 6.x/7.x management station that you are configuring.

5 Click  **Save and Close**.

6 Sign out of the NNMi console.

The next time that you sign in to the NNMi console, the **Actions** menu will contain new items for launching NNM 6.x/7.x views.

Step 3: (Optional) Configure Additional NNM 6.x/7.x Views

The following URLs are not added out-of-the-box. You can add any of these URLs to the NNM 6.x/7.x deployment.

URLs That Do Not Require a Selection

- MIB Browser Example URL:

```
http://192.168.1.xxx:3443/OvCgi/OpenView5.exe?Action=Snmp&Host=speed2.cnd.hp.com
```

- Report Presenter Example URL:

```
http://192.168.1.xxx:3443/OvCgi/nnmRptPresenter.exe
```

- Topology Summary Example URL:

```
http://192.168.1.xxx:7510/topology/summary
```

- SNMP Data Presenter (MIB Form/Table contrib. graphs):

```
http://192.168.1.xxx:3443/OvCgi/snmpviewer.exe?Context=Performance&sel=10.97.245.242
```

- OV Launcher Example URL:

```
http://system.yourcompany.com:3443/OvCgi/ovlaunch.exe
```

- jovw Example URL:

  (Web-based ovw, requires an ovw session running; otherwise, you see the error message `"Cannot find an ovw on host ..."` with map named default using sessionID xxxx:x):

```
http://system.yourcompany.com:3443/OvCgi/jovw.exe
```

This URL can take a context node and map name, with option such as: `jovw.exe?mapName=default&ObjectName=10.1.12.33`

- ovalarm Example URL:

```
http://system.yourcompany.com:3443/OvCgi/ovalarm.exe
```

- Form to request topology details (type in a node by Name, IP Address, Physical Address UUID, OvwId):

```
http://192.168.1.xxx:7510/topology/topoDetail
```

URLs That Require a Selection

- Node Details using an ovwId:

```
http://192.168.1.xxx:7510/topology/
topoDetail?objectType=ovwId&objectValue=3&Show+Details=Show+Details
```

- Node Details using a UUID:

```
http://192.168.1.xxx:7510/topology/
topoDetail?objectType=uuid&objectValue=3dasfasdf&Show+Details=Show+Details
```

Test the Integration

To verify that you have correctly set up the NNM 6.x/7.x integration with the NNMi 8.11 management server, complete one or both of the following procedures, as appropriate:

- Test 1: Verify Event Forwarding
- Test 2: Launch NNM 6.x/7.x Dynamic Views from NNMi 8.11

Test 1: Verify Event Forwarding

Normal NNM 6.x/7.x network monitoring will probably result in the receipt of network events, which will be forwarded to NNMi and displayed as remotely generated 6.x/7.x incidents. To expedite testing, you can generate a test event or create an actual network failure on a test network or device.

To verify event forwarding from the NNM 6.x/7.x management station to the NNMi 8.11 management server, follow these steps:

1. On the NNM 6.x/7.x management station, create a situation that generates one of the forwarded events.

   The simplest approach is to run the `sendMsg.ovpl` command on the NNM 6.x/7.x management station. For information on how to run this command, see sendMsg.ovpl on page 252.

   Another approach is to generate or simulate a network fault on the NNM 6.x/7.x system. See Generate Test Interface Down and Interface Up Events on page 251.

2. In the NNMi 8.11 console, from the workspace navigation panel, select the **Incident Browsing** workspace.

3. Select **NNM 6.x/7.x Events**.

   The event that you generated from the NNM 6.x/7.x management station should be visible in this view.

Alternatively, you can run **`nnmdumpevents -t`** on the NNMi 8.11 management server to see the list of events that the NNMi 8.11 management station has received.

Generate Test Interface Down and Interface Up Events

The following test procedure requires changes to the NNM 6.x/7.x configuration. Do not perform this procedure on a production network management station.

1. On an NNM 7.x management station, disable Extended Topology if it is enabled:

   ```
   setupExtTopo.ovpl -disable
   ```

2. On the NNM 6.x/7.x management station, in the ECS user interface, note which correlations are active, and then disable all correlations.

3 Generate test interface down events, which may also cause a node down event, by running the following command once for each IP interface on the node:

ovtopofix -S Down <*IPADDR*>

Where **<*IPADDR*>** is the IP address of one of the interfaces that has been discovered on the NNM 6.x/7.x management station. To determine the IP addresses to use, run the following command:

ovtopodump > topology.txt

In the `topology.txt` file, search for the word `NODES`, and then locate the entries for the NNM 6.x/7.x management stations. For example:

```
NODES:
  1516        IP    mplscexx.xxx.xx.com    Marginal    10.2.120.72
  1516/1517   IP    mplscexx.xxx.xx.com    Normal      10.2.120.72
  1516/2046   IP    mplscexx.xxx.xx.com    Critical    10.97.255.28
  1516/2047   IP    mplscexx.xxx.xx.com    Critical    10.16.160.5
  1516/2050   -     mplscexx.xxx.xx.com    Normal      -
  1516/2051   -     mplscexx.xxx.xx.com    Normal      -
  1516/2052   -     mplscexx.xxx.xx.com    Normal      -
  1516/2053   -     mplscexx.xxx.xx.com    Normal      -
  1516/5250   IP    mplscexx.xxx.xx.com    Critical    10.40.40.1
  1516/5251   IP    mplscexx.xxx.xx.com    Critical    10.40.40.2
```

When all IP interfaces have status `Critical`, NNM shows the node as down.

Alternatively, you can specify the node name or the topology ID for the NNM 6.x/7.x management station as the last argument to the `ovtopofix` command. For other options, see the *ovtopofix* manpage.

Make sure that the events you are testing (in this case, OV_IF_Up/OV_IF_Down, which are .1.3.6.1.4.1.11.2.17.1.0.58916866 and .1.3.6.1.4.1.11.2.17.1.0.58916867, respectively) are configured to be forwarded to the NNMi 8.11 management server.

4 To clean up the events browser, run the following command once for each IP interface to generate Interface Up and Node Up events:

ovtopofix -S Up <*IPADDR*>

5 On the NNM 6.x/7.x management station, in the ECS user interface, re-enable the correlations that you disabled in step 2.

6 If you disabled Extended Topology in step 1, re-enable it on the NNM 7.x management station:

setupExtTopo.ovpl -enable

sendMsg.ovpl

You can run the `sendMsg.ovpl` command to generate an OV_Message event. For example:

- Windows:

```
%OV_CONTRIB%\NNM\sendMsg\sendMsg.ovpl "" "Test from `hostname` on `date`"
```

- UNIX:

```
$OV_CONTRIB/NNM/sendMsg/sendMsg.ovpl "" "Test from `hostname` on `date`"
```

Each time you run the `sendMsg.ovpl` command, NNM 6.x/7.x generates an OV_Message event containing the text that you included in the `sendMsg.ovpl` command line. For example:

```
1183160690 6 Fri Jun 29 17:44:50 2007 <none>    a Test from
speed2 on Fri Jun 29 17:44:50 MDT 2007;1 17.1.0.58916872 0
```

This event is visible in the **All Alarms** browser on the NNM 6.x/7.x management station.

Best practice

To facilitate identification of new alarms, delete all of the alarms in the **All Alarms** browser before running the `sendMsg.ovpl` command.

By default, the OV_Message incident is not configured in NNMi 8.11. In order to run this test you must have configured forwarding for the OV_Message event in NNM 6.x/7.x, and you must have configured the OV_Message incident in the NNMi 8.11 **Incident Configuration** form.

Test with Traps to NNM 6.x/7.x System

If you configured NNM 6.x/7.x to forward traps, you should see received traps that are being forwarded.

You can manually generate traps on the NNM 6.x/7.x management station with a command similar to the following example:

```
snmptrap -p 162 hostname "" "" 6 1234 "" .1.3.6.1.3.1.1.5.3 \
octetstring "Test Trap"
```

The example generates an SNMP_Link_Down trap. Use the event object identifier for a trap that you configured to be forwarded.

`hostname` is the name of the NNM 6.x/7.x system. For more information, see the *snmptrap* manpage.

Test 2: Launch NNM 6.x/7.x Dynamic Views from NNMi 8.11

1 In the NNMi 8.11 console, open the NNM 6.x/7.x management station that you configured.

The following actions are available on the **Actions** menu:

- NNM 6.x/7.x Home Base
- NNM 6.x/7.x ovw
- NNM 6.x/7.x MIB Browser
- NNM 6.x/7.x Launcher
- NNM 6.x/7.x Alarms

If these actions are not available, sign out of the NNMi console, and then sign in to the NNMi console again.

2 Open each of the views from the **Actions** menu.

Troubleshoot Event Forwarding

If you did not see the expected NNM 6.x/7.x events in the **NNM 6.x/7.x Events** view, follow these steps to troubleshoot the problem:

1 On the NNM 6.x/7.x management station, run the following command:

ovdumpevents -t -l <*n*>

Where **<*n*>** specifies the number of minutes to go back in the event history. For example, when the value for *n* is 1, the `ovdumpevents` command displays the events that have been generated on the NNM 6.x/7.x management station in the last *n* minutes.

2 If an expected event is not included in the `ovdumpevents` output, the event was not generated. Refer to the NNM 6.x/7.x documentation for information about troubleshooting this situation.

3 Repeat step 1 until all expected events are included in the `ovdumpevents` output on the NNM 6.x/7.x management station.

4 On the NNMi 8.11 management server, run the following command:

nnmdumpevents -t -l <*n*>

Where **<*n*>** specifies the number of minutes to go back in the event history. For example, when the value for *n* is 1, the `nnmdumpevents` command displays the events that have been generated on the NNMi 8.11 management server in the last *n* minutes.

5 For each expected event that is not included in the `nnmdumpevents` output, verify the configuration of that event in the **Event Configurator** window on the NNM 6.x/7.x management station.

- Verify that the **Forward Event** option is selected.
- Verify the names or IP addresses of the NNMi 8.11 management servers in the **Forwarded Event Destinations** list.

For more information, see Step 1: Configure NNM 6.x/7.x to Forward Events to the NNMi 8.11 Management Server on page 244.

6 Repeat step 5 until all expected events are included in the `nnmdumpevents` output on the NNMi 8.11 management server.

7 In the NNMi 8.11 console, examine the **NNM 6.x/7.x Events** incident view. If the results are not as expected, verify the incident configuration from the **Remote NNM 6.x/7.x Event Configuration** tab of the **Incident Configuration** form.

Integrations and Plug-ins

This section contains the following chapters:

Using Single Sign-On with NNMi

You can access NNM iSPI applications after signing in to the NNMi console by configuring NNMi's Single Sign-on (SSO) feature. This means that once you sign in to the NNMi console, you automatically have access to other NNM iSPI applications without needing to sign in again. This provides easier access to NNM iSPI applications while maintaining a secure level of NNM iSPI access. Once you sign out of the NNMi console, you will no longer have access to the NNM iSPI application URLs without signing in to them.

SSO Access for NNMi and the iSPIs

To use SSO, you need to access NNMi as follows:

- Use the correct URL in the following form: **http://*<fully_qualified_domain_name>*:*<port_number>*/nnm/** where ***<fully_qualified_domain_name>*** represents the official fully-qualified domain name (FQDN) of the NNMi management server and ***<port_number>*** is the port number assigned during NNMi installation.
- Sign in to NNMi using a valid account.

For SSO to work, URL access to NNMi and NNM iSPIs must share a common network domain name, and not use an IP address in the URL. If you do not have a FQDN for the NNMi server, you can substitute the NNMi server's IP address; however, this will disable the single sign-on for NNM iSPIs, and you will need to sign in again the next time you access the NNM iSPI.

To determine the official FQDN of the NNMi server, use one of the following methods:

- Use the **`nnmofficialfqdn.ovpl`** command to display the value of the official FQDN set during installation. See the *nnmofficialfqdn.ovpl* reference page, or the UNIX manpage, for more information.
- From the NNMi console, click **Help > About Network Node Manager i-series**. Scroll down to find the value for the official FQDN beneath the **Management Server** heading.

If you need to change the official FQDN that was set during installation, use the **`nnmsetofficialfqdn.ovpl`** command. See the *nnmsetofficialfqdn.ovpl* reference page, or the UNIX manpage, for more information.

After installation, the system account is still valid; however, it should only be used for command-line security and for recovery purposes.

SSO to NNM iSPIs requires that users access the NNMi console through a URL that contains the official FQDN. To make it easier for users to satisfy this requirement, you can configure NNMi to redirect NNMi URLs to the official FQDN if a non-official domain name was used, such as an IP address or a shortened version of the domain name. If you do this, it is important that you have an appropriate official FQDN configured. See the NNMi help for more information.

AlarmPoint

AlarmPoint is an interactive alerting application, designed to capture and enrich important events, to route those events to the correct person on any communication device, and to give that person the ability to solve, escalate, or enlist others to resolve the situation.

Through integrations, the AlarmPoint System can become the voice and interface of an automation engine or an intelligent application, such as HP Network Node Manager i Software. When NNMi detects an event that requires attention, AlarmPoint places phone calls or sends pages, instant messages, or e-mail messages to the appropriate personnel, vendors, or customers.

The AlarmPoint System is also persistent, escalating through multiple devices and personnel until someone accepts responsibility or resolves the event. The AlarmPoint System gives the notified person instant, two-way communication with NNMi. Responses are processed immediately on the NNMi management server, enabling remote resolution of the event.

AlarmPoint Express is included with NNMi. For information about purchasing other editions of AlarmPoint software with extended feature sets, contact your HP sales representative.

This chapter contains the following topics:

HP NNMi–AlarmPoint Integration

By including AlarmPoint in the HP Network Node Manager i-series Software environment, network administrators who use NNMi to monitor and manage their network devices gain intelligent alerting and two-way communication between personal communication tools and NNMi.

The HP NNMi–AlarmPoint integration supports event notifications (from NNMi to AlarmPoint) through the configuration of NNMi incident types. It also supports inbound actions (from AlarmPoint to NNMi) to acknowledge the original incident, alter its priority, and add informational annotations.

Value

With the HP NNMi–AlarmPoint integration, the appropriate technician can be notified directly through voice, e-mail, pager, or another device. The event resolver receives information about the failure and is able to make decisions in real time, such as acknowledging, ignoring, annotating, or changing the priority of the event.

After the recipient selects a response on their remote device, AlarmPoint updates the NNMi incident in real time. The benefit is that this process is immediate—significantly faster than the time required for operations staff to notice the failures or malfunctions, determine who is on call, and then manually notify the correct person. Being able to take simple actions to update the incident from any device gives the event resolver a quick way to deal with many issues and communicate to other team members the current status of the incident.

During the process, AlarmPoint logs every notification, response, and action. In addition, AlarmPoint automatically annotates the original NNMi incident with status information.

The AlarmPoint product features a self-service web-based user interface for assigning responsible personnel to each job. AlarmPoint also includes an optional enhanced Subscription panel that allows both managed and self subscription to NNMi incidents.

AlarmPoint Express, a limited edition of AlarmPoint, is available for select versions of NNMi. AlarmPoint Express for HP NNMi has a reduced feature set and comes pre-configured for an NNMi integration. AlarmPoint Express is a great way to get started with the AlarmPoint product family, or it is well-suited for a small production environment that does not require voice or distributed load capability.

Supported Versions

The information in this chapter applies to the following product versions:

- AlarmPoint Java Client version 3.2.1 or higher
- AlarmPoint version 3.2.1 or higher
- NNMi version 8.01 or higher, with an NNMi integration enablement license

Documentation

The HP NNMi–AlarmPoint integration is fully described in the *AlarmPoint for HP Network Node Manager i-series Integration Guide*, which is included with the integration.

The AlarmPoint documentation suite describes the AlarmPoint features and capabilities in detail. The documentation suite is available for download from the AlarmPoint Customer Support Site at:

http://support.alarmpoint.com/

The *AlarmPoint Express for HP NNMi Quick Start Guide* provides an introduction to AlarmPoint's features. This guide describes how to install, configure, and maintain an HP NNMi–AlarmPoint Express integration.

Enabling the HP NNMi–AlarmPoint Integration

The high-level steps for installing and configuring the AlarmPoint for NNMi integration are as follows. For detailed information, refer to the *AlarmPoint for HP Network Node Manager i-series Integration Guide*.

1. Install the AlarmPoint Java Client on the NNMi management server.
2. Install the NNMi-specific integration script for the AlarmPoint Java Client.
3. Install the Web Services Library on the AlarmPoint Webservers and the Application server.
4. *Optional*. Install the AlarmPoint subscription panel for NNMi on the AlarmPoint Webservers.
5. Install the AlarmPoint action scripts for NNMi by using the AlarmPoint Developer IDE.
6. Install the integration voice files to the AlarmPoint Application server.
7. Configure an Event Domain (and, optionally, a Subscription Domain) in AlarmPoint.
8. Configure a Web Services Client for NNMi.
9. Configure the NNMi incident types that should trigger the AlarmPoint scripts.
10. Validate that the integration can inject NNMi incident parameters for AlarmPoint notifications, and that AlarmPoint responses properly update the NNMi incident.

Using the HP NNMi–AlarmPoint Integration

NNMi and AlarmPoint interact by delivering notifications to users and injecting the responses back into NNMi. When NNMi detects a problem in the network (for example, a NonSNMPNodeUnresponsive incident), the following process occurs:

1. NNMi calls the AlarmPoint Client (APClient) with the parameters describing the problem (for example, the computer affected and the situation).
2. The APClient submits the information to the AlarmPoint Agent (APAgent).
3. The APAgent ensures delivery of the problem details to AlarmPoint, which in turn notifies the appropriate recipient.
4. The recipient responds to the notification. The recipient's acknowledgement, annotation, or priority change updates NNMi through a Web services call.

Disabling the HP NNMi–AlarmPoint Integration

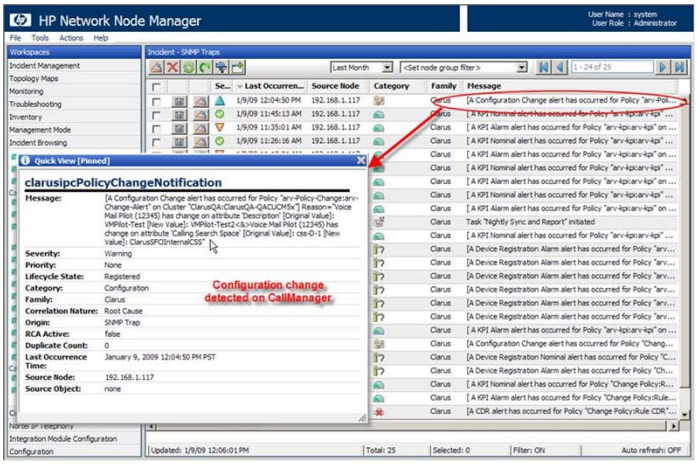
To disable the integration, remove the components installed by the integration's executable archive.

For information about removing an AlarmPoint deployment, refer to the *AlarmPoint for HP Network Node Manager i-series Integration Guide*.

Troubleshooting the HP NNMi–AlarmPoint Integration

For information about optimizing and extending the integration, and any currently known issues, refer to the *AlarmPoint for HP Network Node Manager i-series Integration Guide*.

Clarus Systems ClarusIPC Plus+



Clarus Systems ClarusIPC Plus+ provides voice service testing; remote diagnostics of IP phone features; call detail record (CDR) based alerting and tracking; and reporting of configurations for Cisco Unified Communications Manager IP telephony systems during new deployments, upgrades, and ongoing operations.

Clarus Systems offers an integration of ClarusIPC Plus+ with HP Network Node Manager i Software.
HP offers an integration of ClarusIPC Plus+ with the NNM iSPI for IP Telephony. These integrations are mutually exclusive.

This chapter describes the available integrations:

- HP NNMi–Clarus Systems ClarusIPC Plus+ Integration
- HP NNM iSPI for IP Telephony–Clarus Systems ClarusIPC Plus+ Integration

HP NNMi–Clarus Systems ClarusIPC Plus+ Integration

About the HP NNMi–Clarus Systems ClarusIPC Plus+ Integration

Clarus Systems provides and supports the HP NNMi–Clarus Systems ClarusIPC Plus+ integration. In this integration, ClarusIPC Plus+ forwards SNMP traps regarding IP telephony service test results, alerts based on set CDR policies, or alerts based on the Unified Communications Manager Configuration change policies to NNMi, which then generates incidents regarding the status of the IP telephony configuration and devices. NNMi provides a consolidated view of the entire network.

The integration provides for accessing several ClarusIPC Plus+ tools from these incidents in the NNMi console.

Value

The HP NNMi–Clarus Systems ClarusIPC Plus+ integration consolidates IP telephony device management by providing access from the NNMi console to the ClarusIPC Plus+ tools for IP telephony configuration change tracking and reporting.

Supported Versions

The information in this section applies to the following product versions:

- ClarusIPC Plus+ version 2.6.1 or higher
- NNMi version 8.10 or higher on the Windows operating system only

Documentation

The HP NNMi–Clarus Systems ClarusIPC Plus+ integration is fully described in the *ClarusIPC Plus+ HP NNMi Software Integration Guide*, which is included in the integration installation package.

The ClarusIPC Plus+ documentation suite contains additional documents that describe the ClarusIPC Plus+ features and capabilities in detail. The documentation suite is available for download from the Clarus Systems web site at:

www.support.clarussystems.com

Enabling the HP NNMi–Clarus Systems ClarusIPC Plus+ Integration

To obtain the HP NNMi–Clarus Systems ClarusIPC Plus+ integration installation package, contact Clarus Systems support.

For information about enabling the integration, see the *ClarusIPC Plus+ HP NNMi Software Integration Guide*, which is included in the integration installation package.

Using the HP NNMi–Clarus Systems ClarusIPC Plus+ Integration

Enabling the HP NNMi–Clarus Systems ClarusIPC Plus+ integration adds several URL actions to the NNMi console. For information about these URL actions, see the *ClarusIPC Plus+ HP NNMi Software Integration Guide*.

ClarusIPC Plus+ requires the use of the Microsoft Internet Explorer web browser. Open the NNMi console in Internet Explorer before launching a URL action that opens a ClarusIPC Plus+ window.

Disabling the HP NNMi–Clarus Systems ClarusIPC Plus+ Integration

For information about disabling the HP NNMi–Clarus Systems ClarusIPC Plus+ integration, contact Clarus Systems support.

Troubleshooting the HP NNMi–Clarus Systems ClarusIPC Plus+ Integration

For information about optimizing and extending the integration, and any currently known issues, see the *ClarusIPC Plus+ HP NNMi Software Integration Guide*.

HP NNM iSPI for IP Telephony–Clarus Systems ClarusIPC Plus+ Integration

About the HP NNM iSPI for IP Telephony–Clarus Systems ClarusIPC Plus+ Integration

HP provides and supports the HP NNM iSPI for IP Telephony–Clarus Systems ClarusIPC Plus+ integration. With this integration, operators can access the ClarusIPC Plus+ features pertaining to IP telephony service tests and diagnostics; Cisco Unified Communications Manager Configuration change reports; and CDR monitoring policies. ClarusIPC Plus+ forwards SNMP traps regarding IP telephony service test results, alerts based on set CDR policies, or alerts based on the Unified Communications Manager Configuration change policies to NNMi, which then generates incidents regarding the status of the IP telephony configuration and devices. The NNM iSPI for IP Telephony provides the following:

- Workspaces and menus for launching to ClarusIPC Plus+ configuration change reports, various policies, test plans, and test results
- Launches to ClarusIPC Plus+ remote diagnostics tools for IP phones in the context of the selected IP phone
- Launches to ClarusIPC Plus+ test results, test details, and CDR policy details in the context of the alert incident that is selected in an NNMi incident view.

This integration provides access from the NNMi console to more ClarusIPC Plus+ tools than does the integration without the NNM iSPI for IP Telephony.

Value

The HP NNM iSPI for IP Telephony–Clarus Systems ClarusIPC Plus+ integration adds advanced IP telephony service testing and diagnostics; CDR monitoring; and configuration change tracking and reporting to the NNM iSPI for IP Telephony.

Supported Versions

The information in this chapter applies to the following product versions:

- ClarusIPC Plus+ version 2.6.1 or higher
- NNMi version 8.11 or higher with an NNM iSPI Network Engineering Toolset license

NNMi version 8.11 is a patch to NNMi version 8.10.

- NNM iSPI for IP Telephony version 8.11 or higher, on any supported operating system

NNM iSPI for IP Telephony version 8.11 is a patch to the NNM iSPI for IP Telephony version 8.10.

Documentation

The HP NNM iSPI for IP Telephony–Clarus Systems ClarusIPC Plus+ integration is fully described in the *NNM iSPI for IP Telephony online help*, which is included with the iSPI.

The online help (in PDF format) and additional NNM iSPI for IP Telephony documentation are available at:

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

Enabling the HP NNM iSPI for IP Telephony–Clarus Systems ClarusIPC Plus+ Integration

1. Prepare the NNMi management server:
   a. If the HP NNMi–Clarus Systems ClarusIPC Plus+ integration (provided by Clarus Systems) is installed on the NNMi management server, uninstall that integration before enabling the integration between the NNM iSPI for IP Telephony and ClarusIPC Plus+.

      For information about how to uninstall the ClarusIPC Plus+ integration package, contact Clarus Systems support.
   b. On the NNMi management server, install the following:
      — The most recent NNMi consolidated patch
      — The most recent NNM iSPI for IP Telephony consolidated patch

      Patches are available at:

      http://h20230.www2.hp.com/selfsolve/patches
2. On the NNMi management server, enable the HP NNM iSPI for IP Telephony–Clarus Systems ClarusIPC Plus+ integration as described in the *NNM iSPI for IP Telephony online help*.

Using the HP NNM iSPI for IP Telephony–Clarus Systems ClarusIPC Plus+ Integration

Enabling the HP NNM iSPI for IP Telephony–Clarus Systems ClarusIPC Plus+ integration adds several workspaces, incident types, and URL actions to the NNMi console. For information about these URL actions, see the *NNM iSPI for IP Telephony online help*.

ClarusIPC Plus+ requires the use of the Microsoft Internet Explorer web browser. Open the NNMi console in Internet Explorer before launching a URL action that opens a ClarusIPC Plus+ window.

Disabling the HP NNM iSPI for IP Telephony–Clarus Systems ClarusIPC Plus+ Integration

For information about disabling the HP NNM iSPI for IP Telephony–Clarus Systems ClarusIPC Plus+ integration, see the *NNM iSPI for IP Telephony online help*.

Troubleshooting the HP NNM iSPI for IP Telephony–Clarus Systems ClarusIPC Plus+ Integration

For information about optimizing and extending the integration, and any currently known issues, see the *NNM iSPI for IP Telephony online help*.

For help troubleshooting problems with ClarusIPC Plus+, contact Clarus Systems support.

HP Business Availability Center

HP Business Availability Center (BAC) software provides tool for managing the availability of applications in production, monitoring system performance, monitoring infrastructure performance, and proactively resolving problems when they arise. BAC includes the MyBSM portal for viewing reports and real-time product performance information. (Prior to BAC version 8.0, the portal is called MyBAC.)

This chapter contains the following topics:

HP NNMi–HP BAC Integration

The HP NNMi–HP BAC integration as an enablement for viewing NNMi in the MyBSM portal. The integration provides templates for adding NNMi and NNM iSPI for Performance portlets to a MyBSM portal. For a quick portal demonstration, you can use the configuration tool in the NNMi console to customize these templates for your environment.

The MyBSM administrator can further customize the configuration and access rights of the default portlets by using the standard MyBSM administration interface. The MyBSM administrator can also use the MyBSM administration interface to create custom portlets for other views of NNMi and the NNMi Smart Plug-ins (iSPIs) and to combine views from multiple NNMi management stations on one portal page.

Value

The HP NNMi–HP BAC integration extends the information that can be made available through the MyBSM portal.

Supported Versions

The information in this chapter applies to the following product versions:

- BAC version 7.50 or higher
- NNMi version 8.11 or higher

For the most recent information about supported hardware platforms and operating systems, refer to the support matrices for both products.

Documentation

This chapter describes how to configure the default NNMi portlets for MyBSM from the NNMi console.

The *Using My BAC Guide*, which is included on the BAC 7.x product media, describes how to configure and maintain MyBAC.

The *Using My BSM Guide*, which is included on the BAC 8.x product media, describes how to configure and maintain MyBSM.

Default NNMi Modules for MyBSM

NNMi provides the following MyBSM modules that can be configured in the NNMi console:

- The NNMi demonstration module is defined in the `NOC_Demo_Portal.xml` template file. This module presents some key network status information as described in Table 26.
- The NNMi and NNM iSPI for Performance demonstration module is defined in the `NOC_Demo_Portal_iSPIPerf.xml` template file. This module adds some NNM iSPI for Performance reports to the network status information of the NNMi demonstration module. Table 27 on page 271 describes this module.

Table 26 NNMi Demonstration Module Contents

Page	Portlet	Portlet Description
Overview Map	Key Operations Map	The NNMi topology map for the specified node group.
Network Status/Device Health	Node Group Status	Equivalent to running the **Actions > Status Details** menu command in the NNMi console when a node group is selected.
	Network Status	Equivalent to the Node Groups inventory view in the NNMi console.

Table 27 NNMi and NNM iSPI for Performance Demonstration Module Contents

Page	Portlet	Portlet Description
Overview Map	Key Operations Map	The NNMi topology map for the specified node group.
Network Status/Device Health	Node Group Status	Equivalent to running the **Actions > Status Details** menu command in the NNMi console when a node group is selected.
	Network Status	Equivalent to the Node Groups inventory view in the NNMi console.
	Top-N Memory Utilization	The live NNM iSPI for Performance report for Component Health showing the top 10 nodes by memory utilization.
	Top-N CPU Utilization	The live NNM iSPI for Performance report for Component Health showing the top 10 nodes by CPU utilization.
NNM iSPI for Performance Exceptions	Top-N Devices by Component Exceptions	The live NNM iSPI for Performance dashboard for Component Health showing the top 5 nodes by CPU utilization exceptions and the top 5 nodes by memory utilization exceptions.

Configuring the Demonstration Portlets

This section describes the initial configuration of the out-of-the-box MyBSM modules. The MyBSM administrator can completely customize portlet content and user access to the portlets.

1 On the NNMi management server, create the module configuration XML file:

a In the NNMi console, open the **HP NNMi–HP My BSM Portal Configuration** form (**Integration Module Configuration > HP BAC**).

b Select one of the named XML files to customize:

— If the NNM iSPI for Performance is not installed in your environment, select the `NOC_Demo_Portal.xml` file.

— If the NNM iSPI for Performance is installed in your environment, select the `NOC_Demo_Portal_iSPIPerf.xml` file.

c Click **Load**.

d On each page of the **HP NNMi–HP My BSM Portal Configuration** form, edit the supplied text as appropriate, and then click **Next**. For information about these fields, see HP NNMi—HP My BSM Portal Configuration Form Reference on page 278.

e After navigating through all pages of the **HP NNMi–HP My BSM Portal Configuration** form, click **Finish**, and then save the XML file to a known place on your computer.

f Close the **HP NNMi–HP My BSM Portal Configuration** form.

2 Import the module configuration into BAC:

   a On the BAC **Administration** tab, under **My BSM** (or **My BAC**), click **Import portlets and modules** (one of the options for managing portlet definitions).

   b On the **Import My BSM Objects** page (or the **Import My BAC Objects** page), click **Browse**, and then select the XML file that you saved from the NNMi console.

   c Select the **Replace same Portlet Definitions** check box.

   d Select the **Replace same Modules** check box.

   e Click **Import**.

   The **Import Status** window displays the results of the operation. If the import did not succeed, verify that both check boxes are selected, and then retry the import.

3 View the module in MyBSM or MyBAC:

- Verify that each portlet displays the expected information.
- Use the MyBSM or MyBAC administration tools to define which users have access to the new portlets, to reorganize the pages, to edit the portlet definitions, and so forth.

Creating Custom NNMi Portlets

The easiest way to create a new portlet that displays NNMi or iSPI information is to copy an existing NNMi portlet in the MyBSM administration interface and then change the URL in the portlet definition to point to the information that you want to display in the portal. As you edit the portlet definitions, follow the HTML code structure that is used in the demonstration portlets. For a description of the HTML code structure, see Portlet Definition HTML Reference on page 273.

Determining the Portlet URL

URL for an NNMi console window

For information about how to create a URL for launching an NNMi console window directly, see **Help > NNMi Documentation Library > URL Launch Reference** in the NNMi console.

URL for an NNM iSPI for Performance report

The procedure for determining the URL for launching an NNM iSPI for Performance report depends on which web browser you are using to view the report.

The URL for accessing the report is the same in any web browser.

In Mozilla Firefox, to determine the URL for launching an NNM iSPI for Performance report, follow these steps:

1 Run the NNM iSPI for Performance report.

2 Optional. Click **Show Options** at the top of the report, and then customize the report view.

3 Click **Show URL** at the top of the report.

The URL for the report is visible below the report banner and customization links. You can copy the URL for use in a portlet definition.

4 Optional. Click **Hide URL** to hide the report URL from view.

In Microsoft Internet Explorer, to determine the URL for launching an NNM iSPI for Performance report, follow these steps:

1 Run the NNM iSPI for Performance report.

2 Optional. Click **Show Options** at the top of the report, and then customize the report view.

3 Click **Add Bookmark** at the top of the report.

4 In the **Add a Favorite** window, click **Add**.

5 In the favorites list, right-click the favorite that you created in step 4, and then click **Properties**.

The **URL** field displays the URL for the report. You can copy the URL from this field for use in a portlet definition.

Portlet Definition HTML Reference

The following rules apply to the HTMLPortlet type in BAC:

- Use a single quote character (`'`) instead of the standard double quote character (`"`).
- For every iframe start tag (`<iframe>`) use a corresponding iframe end tag (`</iframe>`) because some web browsers do not recognize empty-element `<iframe/>` tags correctly.
- In the iframe definition, specify one of the following values for `id`:

id Value	Description
nnmi-portlet	The ID of an iframe that displays an NNMi URL. This ID provides consistent configuration for human readers.
nnmi-auth	The ID of an iframe that manages single sign-on to the NNM iSPI for Performance. The JavaScript function for loading the NNM iSPI for Performance portlet interprets this ID value.
ispiperf-portlet	The ID of an iframe that displays an NNM iSPI for Performance report URL. The JavaScript function for loading the NNM iSPI for Performance portlet interprets this ID value.

NNMi portlet HTML structure

A BAC portlet that displays an NNMi URL contains a single iframe element that identifies the NNMi URL to display. The structure is as follows:

```
<html>
<head></head>
<body>
<iframe id='nnmi-portlet' src='<NNMi_URL>'></iframe>
</body>
</html>
```

Replace *<NNMi_URL>* with the URL for launching an NNMi console window.

For example, the following code defines a portlet that displays the status for the Routers node group. In this example, the height and width values are the suggested values for this portlet. You can change them as needed.

```
<html>
<head></head>
<body>
<iframe id='nnmi-portlet'
  src='http://nnmi.example.com:8004/nnm/launch
    ?cmd=runTool
    &tool=nodegroupstatus
    &nodegroup=Routers
    &menus=false'
  width='100%'
  height='425px'>
</iframe>
</body>
</html>
```

NNM iSPI for Performance portlet HTML structure

A BAC portlet that displays an NNM iSPI for Performance report URL contains the following elements:

- Within the portlet header, the declaration of the `loadIspiPerf()` JavaScript function. This declaration defines the NNM iSPI for Performance report to display.
- One iframe element that handles single-sign on from NNMi to the NNM iSPI for Performance.
- A second iframe element that displays the NNM iSPI for Performance report named in the `loadIspiPerf()` function declaration of the portlet header.

The structure of a BAC portlet that displays an NNM iSPI for Performance report URL is as follows:

```
<html>
<head>
<script id='ispiperf-load'>function loadIspiPerf()
{
  var ispiperf_url='<Report_URL>';
  document.getElementById('ispiperf-portlet').src =
    ispiperf_url;
}
</script>
</head>
<body onload='loadIspiPerf();'>
<iframe id='nnmi-auth'
  src='http:<NNMi_host>:<NNMi_port>/nnm/launch?cmd=isRunning'>
</iframe>
```

```
<iframe id='ispiperf-portlet'></iframe>
</body>
</html>
```

Replace *<Report_URL>* with the URL for launching an NNM iSPI for Performance report. Replace *<NNMi_host>* and *<NNMi_port>* with the the fully-qualified domain name of the NNMi management server and the port number for accessing NNMi, respectively.

For example, the following code defines a portlet that displays the node health report for the top N nodes. In this example, the height and width values are the suggested values for this portlet. You can change them as needed.

```
<html>
<head>
<script id='ispiperf-load'>function loadIspiPerf()
{
  var ispiperf_url=
    'http://nnmi.example.com:8004/ssoservlet/protected
    /reports
      ?reportURL=http://ispiperf.example.com:9300/PerfSpi
      /PerfSpi
        ?package=NodeHealth
          &report=Top%20N%20Live
          &element=All%20Nodes/Components&timeperiod=
          &dow=
          &hod=
          &metric=CPU%20Utilization%20(Avg%25)
          &namespaceID=ErsAuthenticationProvider
          &ssoDomain=example.com';
  document.getElementById('ispiperf-portlet').src =
    ispiperf_url;
}
</script>
</head>
<body onload='loadIspiPerf();'>
<iframe id='nnmi-auth'
  src='http://nnmi.example.com:8004/nnm/launch
    ?cmd=isRunning'
  width='100%'
  height='1px'>
</iframe>
<iframe id='ispiperf-portlet'
  width='100%'
  height='750px'>
</iframe>
</body>
</html>
```

Configuring Single Sign-On for the HP NNMi–HP BAC Integration

Single sign-on is available for all HP enterprise applications that use identical initialization string values and share a common network domain name.

If the NNMi and BAC user names are exactly the same for a given individual, that person can log in to the MyBSM portal and view NNMi portlets without also signing in to NNMi. This single sign-on feature maps user names, but not passwords, between the two products. The passwords for signing in to MyBSM and NNMi can be different. Single sign-on does not map user roles, so the user can have different privileges in each application. For example, a user might have normal privileges in BAC and administrator privileges in NNMi.

To configure single sign-on access from BAC to NNMi, make sure that both applications use the same initialization string. You can copy the string from either application to the other. Consider all applications that interact when choosing which initialization string value to use. If necessary, also update the initialization string configuration for other applications or NNM iSPIs.

BAC initialization string

Locate the BAC initialization string as follows:

1. Access the JMX console for BAC at:

   http://*<BAC_hostname>*:*<BAC_JMX_port>*/jmx-console/

2. Select **service=LW-SSO Configuration** (under Topaz).

   The initialization string is the value of the **InitString** parameter.

3. If you change the value of the **InitString** parameter, click **Apply Changes**.

NNMi initialization string

Locate the NNMi initialization string as follows:

1. Open the following file in a text editor:

   - *Windows*: `%NnmInstallDir%\nonOV\jboss\nms\server\nms\conf\lwssofmconf.xml`
   - *UNIX*: `$NnmInstallDir/nonOV/jboss/nms/server/nms/conf/lwssofmconf.xml`

2. Search for the string `initString`.

   The initialization string is the value of the `initString` parameter without the quotation marks.

   For example, if the `lwssofmconf.xml` file contains the following text:

   ```
   initString="E091F3BA8AE47032B3B35F1D40F704B4"
   ```

   the initialization string is:

   ```
   E091F3BA8AE47032B3B35F1D40F704B4
   ```

3. If you change the value of the `initString` parameter, restart `ovjboss`:

   ```
   ovstop ovjboss
   ovstart ovjboss
   ```

Troubleshooting the HP NNMi–HP BAC Integration

The NNMi Portlets Appear As a Sign-in Page

Verify the single sign-on configuration:

1 Sign in to the NNMi console with the user name for logging into MyBSM.

If sign in is not successful, ask the NNMi to configure an account for the MyBSM user.

2 Make sure that BAC and NNMi use the same initialization strings as described in Configuring Single Sign-On for the HP NNMi–HP BAC Integration on page 276.

Also see Single Sign-On Does Not Work Correctly on page 277.

An NNMi Portlet Does Not Load Correctly

In the MyBSM administration interface, verify the NNMi management server host name and port number in the portlet URL.

An NNM iSPI for Performance Portlet Does Not Load Correctly

In the MyBSM administration interface, verify the NNM iSPI for Performance server host name, port number and SSO domain in the portlet URL.

An NNM iSPI for Performance Portlet Displays an AsynchWait_Requests Error

As a BAC portal page loads an NNM iSPI for Performance portlet, it requests information from the NNM iSPI for Performance Cognos database. When a page contains multiple NNM iSPI for Performance portlets, simultaneous requests to the Cognos database from the single web browser session can result in an `AsynchWait_Requests` error. Reload the portal page.

Single Sign-On Does Not Work Correctly

All NNMi and NNM iSPI for Performance portlets do not load. The web browser might close with a message similar to the following:

```
The page you requested cannot be displayed because the LW-SSO
host has logged out.
```

Verify that all applications participating in a LW - SSO integration are set to the same GMT time with a maximum difference of 15 minutes.

MyBSM Reports HTML Validation Errors When I Save A Portlet Definition

Before you can save new portlets in MyBSM, you must configure a setting in a BAC configuration file.

Edit the `<HP Business Availability Center root directory>\HPBAC\conf\dashboard.properties` file to add the following line:

```
Block-URL-Injections=false
```

HP NNMi—HP My BSM Portal Configuration Form Reference

Table 28 lists the fields that are included on the pages of the **HP NNMi–HP My BSM Portal Configuration** form. Coordinate with the NNM iSPI for Performance administrator to determine the correct value for the NNM iSPI for Performance fields. Text fields may contain any characters. NNMi does not validate configured values. Verify the new portlets in the MyBSM portal.

Table 28 Module Configuration Information

Field	Description
Name	The name of the portal module. This text is used for navigation within the portal MyBSM.
Description	Text that describes the portal module. This text is visible in the MyBSM administration interface.
Page Title	The name of a portal page. This text is used for navigation within the portal.
Portlet Title	The name of the portlet as it appears in the portal.
Portlet Type	A field required for the MyBSM configuration. This field is currently read-only.
NNMi Machine	The URL for accessing the NNMi management server. This field is pre-filled with the host name and port number that were used to access the NNMi console. • If this portlet will access the default NNMi management server, leave the default setting. • If this portlet will access a different NNMi management server, manually enter the correct URL.
NNMi Nodegroup	A list of node groups on the NNMi management server that was used to access the NNMi console. • If this portlet will access the default NNMi management server, select a node group from the list. • If this portlet will access a different NNMi management server, manually enter the correct node group name.

Table 28 Module Configuration Information (cont'd)

Field	Description
iSPI for Performance Machine	The URL for accessing the NNM iSPI for Performance server. This field is pre-filled the host name and port number for the NNM iSPI for Performance server that NNMi is configured to use. • If this portlet will access the default NNM iSPI for Performance server, leave the default setting. • If this portlet will access a different NNM iSPI for Performance server, manually enter the correct URL.
SSO Domain	The domain used for single sign-on to the NNM iSPI for Performance.
Disable iSPI for Performance portlets check box	The **Disable iSPI for Performance portlets** check box is present on configuration form pages for defining a portlet that will access the NNM iSPI for Performance. • If this check box is selected when the page first appears, the NNM iSPI for Performance is not configured on this NNMi management station. Therefore, the fields related to the NNM iSPI for Performance are unset. • Clear this check box to enable the fields so that you can manually enter the information for accessing the NNM iSPI for Performance.

HP Network Automation

HP Network Automation software (NA) tracks, regulates, and automates configuration and software changes across globally distributed, multi-vendor networks through process-powered automation.

This chapter contains the following topics:

HP NNMi–HP NA Integration

The HP NNMi–HP NA integration combines NA's configuration change detection capabilities with NNMi's network monitoring capabilities, placing more information at your fingertips when problems occur.

Without exiting NNMi, you can connect to NA to view information about NA-managed devices and configuration change events. While in NA, you can perform any NA functions for which you have the necessary credentials.

The HP NNMi–HP NA integration adds configuration menu items for opening connections to NA. The integration also adds menu items for viewing configuration information on devices managed by NA. These tools provide the following functionality:

- View detailed device information, including vendor, model, modules, operating system version, and recent diagnostic results.
- View device configuration changes and configuration history.

- Compare configurations (typically the most recent and last previous configurations) to see what changed, why, and who made the changes.
- View device compliance information.
- Run NA diagnostics and command scripts from NNMi nodes.
- (NNM iSPI NET) Detect connections with mismatched speed or duplex configurations.

These features are not available for network devices that are not configured in NA or for NA devices for which change detection is disabled.

Value

The HP NNMi–HP NA integration provides the following features and benefits in an environment already running both NNMi and NA:

- Alarm integration—NNMi integration communicates NA configuration change information to the NNMi console, enabling you to quickly identify whether configuration changes may have caused network problems. From within NNMi, you can quickly access NA functionality to view specific configuration changes and device information, identify who made the change, and roll back to the previous configuration to restore network operation. Because a majority of network outages are caused by device configuration errors, this feature can enhance both problem identification and response time in resolving network downtime.
- Access to NA configuration history from NNMi—In the NNMi console, a device-level menu provides access to NA features for reviewing configuration changes. For any device in the NA database, this feature displays configuration changes side-by-side so that you can easily view changes. You can also view configuration history.
- Operations efficiency—Network operations personnel can monitor and investigate information from two data sources within a single screen.

Supported Versions

The information in this chapter applies to the following product versions:

- NA version 7.50 or higher with the latest consolidated patch
- NNMi version 8.11 or higher on the Windows, Linux, or Solaris operating system only

For information about configuring the HP NNMi—HP NA integration for an earlier version of NNMi, see the documentation that is provided with NA.

NNMi and NA can be installed on the same computer or on different computers.

For NNMi and NA to run correctly on the same computer, you must install NNMi before installing NA.

The two products can be installed on different computers in either of the following configurations:

- Different operating systems. For example, the NNMi management server is a Linux system, and the NA server is a Windows system.
- The same operating system. For example, the NNMi management server is a Windows system, and the NA server is a second Windows system.

For the most recent information about supported hardware platforms and operating systems, refer to the support matrices for both products.

Documentation

This chapter describes how to configure NNMi to communicate with NA and how to use the integration from the NNMi console.

The *HP Network Automation NNM Integration User's Guide*, which is included on the NA product media, describes how to configure NA to communicate with NNMi. It also describes how to use the integration from the NA user interface.

Enabling the HP NNMi–HP NA Integration

To enable the HP NNMi—HP NA integration for NNMi version 8.11 or higher, follow these steps:

The January 2009 edition of the *HP Network Automation NNM Integration User's Guide* for NA version 7.50 describes how to configure the integration with NNMi version 8.10. For NNMi version 8.11 or higher, the instructions in this chapter supersede those in the *HP Network Automation NNM Integration User's Guide*.

1. On the NA server, install the NNMi connector that is provided in the latest consolidated NA patch:
   - If the NA server is a different computer from the NNMi management server, run one of the following connector scripts. Choose the connector that corresponds to the operating system of the NA server:
     - `na_nnm_connector_windows.exe`
     - `na_nnm_connector_solaris.bin`
     - `na_nnm_connector_linux.bin`
   - If the NA server is the same computer as the NNMi management server, run one of the following connector scripts. Choose the connector that corresponds to the operating system of the NA server:
     - `na_nnm_coresidency_windows.exe`
     - `na_nnm_coresidency_solaris.bin`
     - `na_nnm_coresidency_linux.bin`

The connector scripts detects the NNMi management server and installs the components for communicating with NA on the NNMi management server. The connector script also imports the NNMi topology into the NA database.

When the connector script asks for the NNMi HTTP Port, enter the port for connecting to the NNMi console. This port is set during NNMi installation and is specified in the following file:

- *Windows*: `%NnmDataDir%\shared\nnm\conf\nnm.ports.properties`
- *UNIX*: `$NnmDataDir/shared/nnm/conf/nnm.ports.properties`

For non-SSL connections, use the value of `jboss.http.port`, 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 `jboss.https.port`, which is `443` by default.

2 On the NNMi management server, configure the connection between NNMi and NA:

a In the NNMi console, open the **HP NNMi–HP NA Integration Configuration** form (**Integration Module Configuration > HP NA**).

b Select the **Enable Integration** check box to activate the remaining fields in the form.

c Enter the information for connecting to the NNMi management server. For information about these fields, see NNMi Management Server Connection on page 291.

d Enter the information for connecting to the NA server. For information about these fields, see NA Server Connection on page 292.

The integration requires an http connection to the NA web services. Leave the **HP NA SSL Enabled** check box cleared.

e (NNMi Advanced) Enter a value in the **HP NA Connection Check Interval** field. For information about this field, see Integration Behavior on page 292.

f Click **Submit** at the bottom of the form.

A new window displays a status message. If the message indicates a problem with connecting to the NA server, re-open the **HP NNMi–HP NA Integration Configuration** form (or press **ALT+LEFT ARROW** in the message window), and then adjust the values for connecting to the NA server as suggested by the text of the error message.

Using the HP NNMi–HP NA Integration

The HP NNMi–HP NA integration provides links to NA from the NNMi console. The integration does not provide single sign-on between the products. You must enter your NA user credentials to view the NA windows.

Enabling the HP NNMi–HP NA integration adds the following URL actions to the NNMi console:

- **Show HP NA Diagnostic Results**—Displays a list of the NA tasks that have been scheduled for an NNMi incident. Select a task to view the task results. For more information, see Viewing the Results of Incident Actions that Access NA on page 286.
- **Rerun HP NA Diagnostics**—Runs any NA actions that are configured for the incident. For more information, see Viewing the Results of Incident Actions that Access NA on page 286.
- (NNM iSPI NET) **Show mismatched connections**—Displays a table of all layer 2 connections with possible speed or duplex configuration differences. For more information, see Identifying Layer 2 Connections with Mismatched States (NNM iSPI NET) on page 287.
- **View HP NA Device Information**—Opens the current **Device Details** page for the selected device in NA.
- **View HP NA Device Configuration**—Opens the **Current Configuration** page for the selected device in NA.

  If real-time change detection is disabled for any device, the most recent configuration will be the configuration captured by NA at the last device polling interval. If configuration changes were made following that interval, this may not be the current configuration.

- **View HP NA Device Configuration Diffs**—Opens the **Compare Device Configuration** page for the selected device in NA.
- **View HP NA Device Configuration History**—Opens the **NA Device Configurations History** page for the selected device in NA.
- **View HP NA Policy Compliance Report**—Opens the **Policy, Rule and Compliance Search Results** page for the selected device in NA.
- **Telnet to HP NA Device**—Opens a **Telnet** window for connecting to the selected device in NA.
- **SSH to HP NA Device**—Opens an **SSH** window for connecting to the selected device in NA.
- **Launch HP NA**—Opens the NA user interface.
- **Launch HP NA Command Scripts**—Opens the **New Task—Run Command Script** page in NA. The page is pre-filled for the node or incident that was selected in the NNMi console.
- **Launch HP NA Diagnostics**—Opens the **New Task—Run Diagnostics** page in NA. The page is pre-filled for the node or incident that was selected in the NNMi console.

For information about using the NA functionality, see the *HP Network Automation User's Guide*.

Configuring NA Diagnostics and Command Scripts as Incident Actions

Enabling the HP NNMi–HP NA integration modifies some out-of-the-box NNMi incidents to include incident actions that access NA each time the associated incident type occurs. Table 29 lists the modified incidents.

Table 29 NNMi Incidents Configured with NA Actions

NNMi Incident	NA Actions Added to the Incident
OSPFNbrStateChange	Show Neighbor
OSPFVirtIfStateChange	Show Neighbor
OSPFIfStateChange	Show Neighbor Show Interfaces
InterfaceDown	Show Interfaces
CiscoChassisChangeNotification	Show Module

You can add an action that accesses NA to any other NNMi incident, and you can modify the default incident actions. On the **Action Configuration** tab for an incident, add a new lifecycle transition action with **Command Type** of `ScriptOrExecutable`. In the **Command** field, enter either `naruncmdscript.ovpl` or `narundiagnostic.ovpl` with the appropriate arguments. For examples, see the action configurations of the incidents listed in Table 29.

Viewing the Results of Incident Actions that Access NA

When an incident of a type that has been configured with an NA action arrives, NNMi initiates the configured action and stores the task ID of the diagnostic or command script as an attribute of that incident. The presence of the task ID enables the **Show HP NA Diagnostic Results** and **Rerun HP NA Diagnostics** items on the **Actions** menu.

To view the outcome of the action at the time the incident occurred, select the incident in any incident view, and then select **Actions > Show HP NA Diagnostic Results**.

To view current results of the configured action, select the incident in any incident view, and then select **Actions > Rerun HP NA Diagnostics**.

If you run the task multiple times, NNMi lists the most recent task ID on the **Custom Attributes** tab. The **Show HP NA Diagnostic Results** action displays all of the tasks that have been run for the incident so that you can compare the results from different runs.

Identifying Layer 2 Connections with Mismatched States (NNM iSPI NET)

If an NNM iSPI NET license key is installed on the NNMi management server and the HP NNMi—HP NA integration is enabled, NNMi periodically queries NA for the speed and duplex settings of the two interfaces on either end of each layer 2 connection in the NNMi topology. Additionally, NNMi queries NA for the speed and duplex settings of the interfaces for any new connection that is added to the NNMi topology and, when the NNM iSPI for Performance is running, for any connection with performance threshold exceptions that might indicate a mismatched connection. NNMi uses a mismatch detection algorithm to determine whether the values may result in a mismatched connection.

NNMi is able to perform the mismatch analysis only when the NA inventory includes the MAC addresses for both interfaces that form a layer 2 connection. If the NA interface records do not include valid MAC addresses, run the NA **Topology Data Gathering** diagnostic to update the MAC address fields.

The **Actions->Show mismatched connections** menu item displays a table of layer 2 connections that NNMi suspects may contain speed and/or duplex mismatches. For each suspect connection, the table lists the speed and duplex values for the interfaces on either side of the connection and NNMi's interpretation of the data. The possible interpretations are as follows:

- `MATCH` indicates that the speed and duplex values most likely result in a properly functioning layer 2 connection.
- `POSSIBLE_MISMATCH` indicates that the speed, duplex, or both values may potentially conflict and result in a poor or non-performing connection.
- `MISMATCH` indicates that the speed, duplex, or both values most likely conflict and result in a poor or non-performing connection.

The **HP NA Connection Check Interval** parameter on the **HP NNMi–HP NA Integration Configuration** form specifies the frequency of the connection queries.

Importing NNMi 8.10 Devices into the NA Inventory

The NA inventory does not automatically update as the NNMi topology changes. After adding new devices to the NNMi topology, import those devices into the NA inventory.

To import NNMi device information into the NA inventory, follow these steps:

1. On the NNMi management server, change to the NA root directory. The default location is:
   - *Windows*: `C:\NA`
   - *UNIX*: `/opt/NA`
2. Run the following command:
   - *Windows*: `nnmimport.bat`
   - *UNIX*: `nnmimport.sh`

Changing the HP NNMi–HP NA Integration

1. In the NNMi console, open the **HP NNMi–HP NA Integration Configuration** form (**Integration Module Configuration > HP NA**).
2. Modify the values as appropriate. For information about the fields on this form, see HP NNMi–HP NA Integration Configuration Form Reference on page 290.
3. Verify that the **Enable Integration** check box at the top of the form is selected, and then click **Submit** at the bottom of the form.

The changes take effect immediately. You do not need to restart `ovjboss`.

Disabling the HP NNMi–HP NA Integration

1. In the NNMi console, open the **HP NNMi–HP NA Integration Configuration** form (**Integration Module Configuration > HP NA**).
2. Clear the **Enable Integration** check box at the top of the form, and then click **Submit** at the bottom of the form. The integration URL actions are no longer available.

The changes take effect immediately. You do not need to restart `ovjboss`.

Troubleshooting the HP NNMi–HP NA Integration

If the integration has worked successfully in the past, it is possible that some aspect of the configuration, for example, the NNMi or NA user password, has changed recently. You may want to update the integration configuration as described in HP NNMi–HP NA Integration Configuration Form Reference on page 290, before walking through this entire procedure.

1. In the NNMi console, open the **HP NNMi–NA Integration Configuration** form (**Integration Module Configuration > HP NA**).

   For information about the fields on this form, see HP NNMi–HP NA Integration Configuration Form Reference on page 290.
2. To check the status of the integration, in the **HP NNMi–HP NA Integration Configuration** form, click **Submit** at the bottom of the form (without making any configuration changes).

   A new window displays a status message.

   If the message indicates a problem with connecting to the NA server, NNMi and NA are not able to communicate. Continue with step 3 of this procedure.

3 To verify the accuracy and access level of the NA credentials, log in to the NA user interface with the credentials for the **HP NA User** from the **HP NNMi–HP NA Integration Configuration** form.

If you cannot log in to the NA user interface, contact the NA administrator to verify your log in credentials.

4 To verify that the connection to the NA server is configured correctly, in a web browser on the NNMi management server, enter the following URL:

http://<*naserver*>:<*naport*>/soap

Where the variables are related to values on the **HP NNMi–HP NA Integration Configuration** form as follows:

- The **HP NA SSL Enabled** check box must be cleared to indicate that connections to the NA server use the `http` protocol.
- *<naserver>* is the value of **HP NA Host**.
- *<naport>* is the value of **HP NA Port**.

If the NA web service is running on the specified server and port, then the NA server responds with the following message:

```
NAS SOAP API: Only handles HTTP POST requests
```

- If the expected message appears, the connection to the NA server (server name, port, and protocol) is configured correctly. Continue with step 5.
- If you see an error message, the connection to the NA server is not configured correctly. Contact the NA administrator to verify the information that you are using to connect to the NA web services. Continue to troubleshoot the connection to NA until you see the expected message.

5 Verify that the connection to NNMi is configured correctly:

If you used the information described in this step to connect to the NNMi console in step 1 of this procedure, you do not need to reconnect to the NNMi console. Continue with step 6.

a In a web browser, enter the following URL:

***<protocol>://<NNMiserver>:<port>*/nnm/**

Where the variables are related to values on the **HP NNMi–HP NA Integration Configuration** form as follows:

— If the **NNMi SSL Enabled** check box is selected, `<protocol>` is `https`.

— If the **NNMi SSL Enabled** check box is cleared, `<protocol>` is `http`.

— `<NNMiserver>` is the value of **NNMi Host**.

— `<port>` is the value of **NNMi Port**.

b When prompted, enter the credentials for an NNMi user with the Administrator role.

You should see the NNMi console. If the NNMi console does not appear, contact the NNMi administrator to verify the information that you are using to connect to NNMi. Continue to troubleshoot the connection to NNMi until the NNMi console appears.

You cannot sign into the NNMi console as a user with the Web Service Client role.

c Verify the values of the **NNMi User** and **NNMi Password**.

— If the **NNMi User** listed on the **HP NNMi–HP NA Integration Configuration** form has the Administrator role and you were able to connect to the NNMi console with this user name, then re-enter the corresponding password on the **HP NNMi–HP NA Integration Configuration** form.

— If the **NNMi User** listed on the **HP NNMi–HP NA Integration Configuration** form has the Web Service Client role, contact the NNMi administrator to verify the values of **NNMi User** and **NNMi Password**.

Passwords are hidden in the NNMi console. If you are not sure what password to specify for an NNMi user name, ask the NNMi administrator to reset the password.

6 Update the **HP NNMi–HP NA Integration Configuration** form with the values that you used for successful connections in step 4 and step 5 of this procedure.

For more information, see HP NNMi–HP NA Integration Configuration Form Reference on page 290.

7 Click **Submit** at the bottom of the form.

8 If the status message still indicates a problem with connecting to the NA server, do the following:

a Clear the web browser cache.

b Clear all saved form or password data from the web browser.

c Close the web browser window completely, and then re-open it.

d Repeat step 6 and step 7 of this procedure.

9 Test the configuration by launching one of the URL actions listed in Using the HP NNMi–HP NA Integration on page 285.

HP NNMi–HP NA Integration Configuration Form Reference

The **HP NNMi–HP NA Integration Configuration** form contains the parameters for communications between NNMi and NA. This form is available from the **Integration Module Configuration** workspace.

Only NNMi users with the Administrator role can access the **HP NNMi–HP NA Integration Configuration** form.

The **HP NNMi–HP NA Integration Configuration** form collects information for the following general areas:

- NNMi Management Server Connection
- NA Server Connection
- Integration Behavior

To apply changes to the integration configuration, update the values on the **HP NNMi–HP NA Integration Configuration** form, and then click **Submit**.

NNMi Management Server Connection

Table 30 lists the parameters for connecting to the NNMi management server. This is the same information that you use to open the NNMi console. You can determine many of these values by examining the URL that invokes an NNMi console session. Coordinate with the NNMi administrator to determine the appropriate values for this section of the configuration form.

The default NNMi configuration uses http for connecting to the NNMi console. For information about configuring this connection to use https, see Enabling https for NNMi on page 77.

Table 30 NNMi Management Server Information

Field	Description
HP NNMi SSL Enabled	The connection protocol specification. • If the NNMi console is configured to use https, select the **NNMi SSL Enabled** check box. • If the NNMi console is configured to use http, clear the **NNMi SSL Enabled** check box. This is the default configuration.
HP 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. Verify that this value is the name that is returned by the **`nnmofficialfqdn.ovpl -t`** command run on the NNMi management server.
HP NNMi Port	The port for connecting to the NNMi console. This field is pre-filled with the port that the jboss application server uses for communicating with the NNMi console, as specified in the following file: • *Windows*: `%NnmDataDir%\shared\nnm\conf\nnm.ports.properties` • *UNIX*: `$NnmDataDir/shared/nnm/conf/nnm.ports.properties` For non-SSL connections, use the value of `jboss.http.port`, 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 `jboss.https.port`, which is `443` by default.
HP NNMi User	The user name for connecting to the NNMi console. This user must have the NNMi Administrator or Web Service Client role.
HP NNMi Password	The password for the specified NNMi user.

NA Server Connection

Table 31 lists the parameters for connecting to the web services on the NA server. Coordinate with the NA administrator to determine the appropriate values for this section of the configuration.

Table 31 NA Server Information

HP NA Server Parameter	Description
HP NA SSL Enabled	The connection protocol specification for connecting to the NA web services. The integration requires an http connection to the NA web services. Leave the **HP NA SSL Enabled** check box cleared.
HP NA Host	The fully-qualified domain name or the IP address of the NA server.
HP NA Port	The port for connecting to the NA web services. The default NA ports are as follows: • 80—for connections to NA on a separate computer from NNMi • 8080—for connections to NA on the same computer as NNMi
HP NA User	A valid NA user account name with the NA Administrator role.
HP NA Password	The password for the specified NA user.

Integration Behavior

The **HP NA Connection Check Interval** parameter specifies the frequency with which NNMi verifies with NA the interface data for all layer 2 connections in the NNMi topology as described in Identifying Layer 2 Connections with Mismatched States (NNM iSPI NET) on page 287. The default interval for the connection check is 24 hours.

The connection check functionality requires an NNM iSPI NET license. If an NNM iSPI NET license key is not installed on the NNMi management server, the **HP NA Connection Check Interval** field is disabled.

HP Operations Manager

HP Operations Manager (HPOM) provides comprehensive event management; proactive performance monitoring; and automated alerting, reporting, and graphing for management operating systems, middleware, and application infrastructure. HPOM consolidates events from a wide range of sources into a single view.

For information about purchasing HPOM, contact your HP sales representative.

This chapter contains the following topics:

HP NNMi–HPOM Integration

The HP NNMi–HPOM integration forwards NNMi incidents to the HPOM active messages browser. The integration synchronizes incidents between NNMi and HPOM. It also provides for accessing the NNMi console from within HPOM.

The HP NNMi–HPOM integration supports a “many-to-many” arrangement. Each NNMi management server can forward incidents to multiple HPOM management servers. Likewise, each HPOM management server can receive incidents from multiple NNMi management servers. The integration interprets the unique identifier of an incident to determine the source NNMi management server.

The HP NNMi–HPOM integration consists of the following components:

- **HP NNMi–HPOM Integration Module**

  The HP NNMi–HPOM integration module forwards incidents from NNMi to HPOM. It is installed and configured on the NNMi management server.

- **HP Operations Manager Incident Web Service**

  HPOM uses the HP Operations Manager Incident Web Service (IWS) to receive the incidents that are forwarded from NNMi.

- **HPOM applications for contextual access of the NNMi console**

  HPOM provides applications for accessing forms, views, and tools in the NNMi console. For example, you can open an NNMi incident directly from the HPOM active messages browser. The specific application determines the context in which the NNMi console opens. You need to configure the applications before you can use them.

Value

The HP NNMi–HPOM integration provides event consolidation in the HPOM active messages browser for the network management, system management, and application management domains, so that HPOM users can detect and investigate potential network problems.

The primary features of the integration are as follows:

- Automatic incident forwarding from NNMi to HPOM.
  - Forwarded incidents appear in the HPOM active messages browser.
  - You can create filters that limit which incidents NNMi forwards.
- Synchronization of Incident updates between NNMi and HPOM as described in the following table.

Trigger	Result
In HPOM, the message is acknowledged.	In NNMi, the corresponding incident's lifecycle state is set to Closed.
In HPOM, the message is unacknowledged.	In NNMi, the corresponding incident's lifecycle state is set to Registered.
In NNMi, the incident's lifecycle state is set to Closed.	In HPOM, the corresponding message is acknowledged.
In NNMi, the incident's lifecycle state is changed from Closed to any other state.	In HPOM, the corresponding message is unacknowledged.

- Access to the NNMi console from HPOM.
  - HPOM users can open the NNMi Incident form in the context of a selected message.
  - HPOM users can launch an NNMi view (for example, the Layer 2 Neighbor view) in the context of a selected message and node.

— HPOM users can launch an NNMi tool (for example, status poll) in the context of a selected message and node.

— When HPOM is consolidating NNMi incidents from multiple NNMi management servers, the integration interprets the unique identifier of each incident to access the correct NNMi management server.

Supported Versions

The information in this chapter applies to the following product versions:

- HPOM for Windows version 8.10 or higher
- HPOM for UNIX version 8.30 or higher
- NNMi version 8.10 or higher

For information about how to configure the HP NNMi–HPOM for NNMi 8.0x (starting with NNMi 8.03), see the chapter *Integration with HP Operations Manager* in the *NNMi Deployment Guide* that corresponds to your version of NNMi.

NNMi and HPOM cannot be installed on the same computer. The two products must be installed on different computers in either of the following configurations:

- Different operating systems. For example, the NNMi management server is a Linux system, and the HPOM management server is a Windows system.
- The same operating system. For example, the NNMi management server is a Windows system, and the HPOM management server is a second Windows system.

For the most recent information about supported hardware platforms and operating systems, refer to the support matrices for both products.

Documentation

This chapter describes how to configure NNMi to communicate with HPOM.

The HPOM documentation describes how to configure HPOM to communicate with NNMi. It also describes how to use the HP NNMi–HPOM integration.

- For HPOM for Windows, see the information for the HP NNMi Adapter in the HPOM online help.
- For HPOM for UNIX, see the *HP NNMi–HPOM Integration for HP Operations Manager User's Guide*.

Enabling the HP NNMi–HPOM Integration

This section describes the procedure for enabling the HP NNMi–HPOM integration. For each NNMi management server and each HPOM management server that you want to include in the integration, complete the appropriate steps in the procedure for the version of HPOM that you are using.

HPOM for Windows

1. On the NNMi management server, configure NNMi incident forwarding to HPOM:
   a. In the NNMi console, open the **HP NNMi–HPOM Integration Configuration** form (**Integration Module Configuration > HPOM**).
   b. Select the **Enable Integration** check box to activate the remaining fields in the form.
   c. Enter the information for connecting to the NNMi management server.

      For information about these fields, see NNMi Management Server Connection on page 305.
   d. Enter the information for connecting to the HPOM management server.

      For information about these fields, see HPOM Management Server Connection on page 307.
   e. If you want NNMi to forward incidents to multiple HPOM management servers, click **Add another OM server**, and then enter the information for the next HPOM management server in the HPOM fields.

      The information for the first server appears in the **Additional HPOM Servers** list.
   f. Click **Submit** at the bottom of the form.

      A new window displays a status message. If the message indicates a problem with connecting to the HPOM server, re-open the **HP NNMi–HPOM Integration Configuration** form (or press **ALT+LEFT ARROW** in the message window), and then adjust the values for connecting to the HPOM management server as suggested by the text of the error message.
2. On the NNMi management server, customize the integration:
   a. In the NNMi console, open the **HP NNMi–HPOM Integration Configuration** form (**Integration Module Configuration > HPOM**).
   b. Enter values for the following fields:
      - **Forward Only**
      - **Holding period (minutes)**
      - **Incident Filter**

      For information about these fields, see Integration Behavior on page 308.
   c. Click **Submit** at the bottom of the form.
3. In HPOM, configure the NNMi adapter for connecting to the NNMi management server as described in *Configure the NNMi Server Name and Port* of the HPOM for Windows online help.

4 In HPOM, create a managed node for each NNMi node that will be named as a source node in the NNMi incidents that are forwarded to this HPOM management server. Also create a managed node for each NNMi management server that will forward incidents to this HPOM management server.

Alternatively, you can create one external node to catch all forwarded NNMi incidents.

For more information, see *Configuring NNMi Server Nodes* in the HPOM online help.

If you do not set up an HPOM managed node for an NNMi incident source node, the HPOM management server discards all incidents regarding that node.

5 In HPOM, add the custom message attributes for NNMi incidents to the active messages browser:

a In the browser, right-click any column heading, and then click **Options**.

b In the **Enter Custom Message Attributes** list, select an attribute, and then click **Add**.

— The custom message attributes for NNMi incidents begin with the text `nnm`.

— The most interesting attributes for NNMi incidents are as follows:

```
nnm.assignedTo
nnm.category
nnm.emittingNode.name
nnm.source.name
```

— To change the order in which the custom message attributes appear in the messages browser, drag a column heading to the new location.

6 In HPOM, enable contextual launching of the NNMi views by associating the NNMi source nodes with the HP NNMi Web Tools group.

For more information, see *Enabling HP NNMi Web Tools in the By Node Group* in the HPOM online help.

HPOM for UNIX

1 Prepare the HPOM for UNIX management server:

a On the HPOM for UNIX management server, install the HP Operations Manager Incident Web Service (IWS) as described in the *HP Operations Manager Incident Web Service Integration Guide*.

b On the HPOM for UNIX management server, install the most recent HPOM consolidated patch, which is available at:

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

2 On the NNMi management server, configure NNMi incident forwarding to HPOM:

a In the NNMi console, open the **HP NNMi–HPOM Integration Configuration** form (**Integration Module Configuration > HPOM**).

b Select the **Enable Integration** check box to activate the rest of the fields in the form.

c Enter the information for connecting to the NNMi management server.

For information about these fields, see NNMi Management Server Connection on page 305.

d Enter the information for connecting to the HPOM management server.

For information about these fields, see HPOM Management Server Connection on page 307.

e If you want NNMi to forward incidents to multiple HPOM management servers, click **Add another OM server**, and then enter the information for the next HPOM management server in the HPOM fields.

The information for the first server appears in the **Additional HPOM Servers** list.

f Click **Submit** at the bottom of the form.

A new window displays a status message. If the message indicates a problem with connecting to the HPOM server, re-open the **HP NNMi–HPOM Integration Configuration** form (or press **ALT+LEFT ARROW** in the message window), and then adjust the values for connecting to the HPOM management server as suggested by the text of the error message.

3 On the NNMi management server, customize the integration:

a In the NNMi console, open the **HP NNMi–HPOM Integration Configuration** form (**Integration Module Configuration > HPOM**).

b Enter values for the following fields:

— **Forward Only**

— **Holding period (minutes)**

— **Incident Filter**

For information about these fields, see Integration Behavior on page 308.

c Click **Submit** at the bottom of the form.

4 In HPOM, create a managed node for each NNMi node that will be named as a source node in the NNMi incidents that are forwarded to this HPOM management server. Also create a managed node for each NNMi management server that will forward incidents to this HPOM management server.

Alternatively, you can create one external node to catch all forwarded NNMi incidents.

For more information, see the *HP Operations Manager for UNIX Administrator's Reference*.

If you do not set up an HPOM managed node for an NNMi incident source node, the HPOM management server discards all incidents regarding that node.

5 In HPOM, add the custom message attributes for NNMi incidents to the active messages browser:

a In the browser, right-click any column heading, and then click **Customize Message Browser Columns**.

b On the **Custom** tab, select from the **Available Custom Message Attributes**, and then click **OK**.

— The custom message attributes for NNMi incidents begin with the text `nnm`.

— The most interesting attributes for NNMi incidents are as follows:

```
nnm.assignedTo
nnm.category
nnm.emittingNode.name
nnm.source.name
```

— To change the order in which the custom message attributes appear in the messages browser, drag a column heading to the new location.

6 On the HPOM management server, prepare the HPOM applications for accessing the NNMi console.

a *Required*. Install the basic set of NNMi applications.

b *Optional*. Install additional NNMi applications.

For more information, see the chapter on installing and configuring the HP NNMi–HPOM integration in the *HP NNMi–HPOM Integration for HP Operations Manager User's Guide*.

Using the HP NNMi–HPOM Integration

Usage Example

Figure 17 shows an address not responding incident in the NNMi console. The information in the **Source Object** and **Message** columns together describe the situation.

Figure 17 Interface Down Incident in NNMi Console

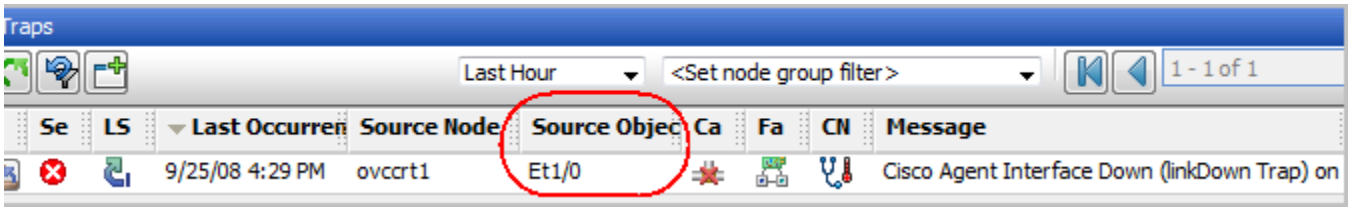


Figure 18 shows the NNMi incident as received by HPOM for Windows. Figure 19 shows the NNMi incident as received by HPOM for UNIX. The **nnm.source.name** and **Text** columns are equivalent to the **Source Object** and **Message** columns in the NNMi console.

You must enable the display of the **nnm.source.name** custom message attribute column as described in step 5 on page 297 (for HPOM for Windows) and in step 5 on page 299 (for HPOM for UNIX).

Figure 18 Forwarded Incident in HPOM for Windows

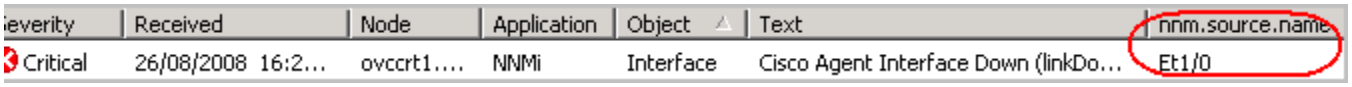


Figure 19 Forwarded Incident in HPOM for UNIX

A Normal Situation: Unknown MSI Condition

The HPOM server receives forwarded NNMi incidents through MSI (not a regular trap template). In the HPOM message browser, the format of the message source is **MSI** followed by the name of the MSI interface. The condition name corresponds to the `condition_id` field in the message, which is unset because there is no template.

- In HPOM for Windows, the policy type is empty.
- In HPOM for UNIX, the message source is of the format:
  MSI: <*MSI_Interface*>: Unknown Condition.

More Information

For detailed information about using the HP NNMi–HPOM integration, see the HPOM documentation.

- For HPOM for Windows, see *NNMi Web Tools* in the HPOM online help.
- For HPOM for UNIX, see the chapter on using the NNMi applications in the *HP NNMi–HPOM Integration for HP Operations Manager User's Guide*.

The NNMi help does not describe the new functionality that this integration adds to the NNMi console. Provide each NNMi operator with the integration information that is pertinent to that operator's job responsibilities.

In the HPOM messages browser, the details for a forwarded NNMi incident are available as custom message attributes.

Changing the HP NNMi–HPOM Integration Configuration

1. In the NNMi console, open the **HP NNMi–HPOM Integration Configuration** form (**Integration Module Configuration > HPOM**).
2. Modify the values as appropriate.
   - If you know the syntax of the entries in the Incident Filter and Additional HPOM Servers lists, you can modify the entries directly.
   - If you do not know the syntax for a list item, delete that entry and then re-enter it.

For information about the fields on this form, see HP NNMi–HPOM Integration Configuration Form Reference on page 305.

3 Verify that the **Enable Integration** check box at the top of the form is selected, and then click **Submit** at the bottom of the form.

The changes take effect immediately. You do not need to restart `ovjboss`.

Disabling the HP NNMi–HPOM Integration

For All HPOM Management Servers

To discontinue the forwarding of NNMi incidents to all HPOM management servers, follow these steps:

1 In the NNMi console, open the **HP NNMi–HPOM Integration Configuration** form (**Integration Module Configuration > HPOM**).

2 Clear the **Enable Integration** check box at the top of the form, and then click **Submit** at the bottom of the form.

The changes take effect immediately. You do not need to restart `ovjboss`.

If necessary, repeat this process for all NNMi management servers.

For One HPOM Management Server

To discontinue the forwarding of NNMi incidents to only one of the HPOM management servers, follow these steps:

1 In the NNMi console, open the **HP NNMi–HPOM Integration Configuration** form (**Integration Module Configuration > HPOM**).

2 In the **Additional HPOM Servers** list, edit the text to delete the entry (or entries) for the HPOM management server to disconnect from the integration.

Clicking **Clear** removes all HPOM servers from the list.

3 Click **Submit** at the bottom of the form.

The changes take effect immediately. You do not need to restart `ovjboss`.

Troubleshooting the HP NNMi–HPOM Integration

HPOM Does Not Receive Any Forwarded Incidents

If the integration has worked successfully in the past, it is possible that some aspect of the configuration, for example, the NNMi or HPOM user password, has changed recently. You may want to update the integration configuration as described in Changing the HP NNMi–HPOM Integration Configuration on page 300, before walking through this entire procedure.

1. In the NNMi console, open the **HP NNMi–HPOM Integration Configuration** form (**Integration Module Configuration > HPOM**).

   For information about the fields on this form, see HP NNMi–HPOM Integration Configuration Form Reference on page 305.

2. To check the status of the integration, in the **HP NNMi–HPOM Integration Configuration** form, click **Submit** at the bottom of the form (without making any configuration changes).

   A new window displays a status message.

   - If the message indicates success, the problem is most likely that HPOM is not configured to accept incidents from the devices that NNMi manages. HPOM ignores any forwarded incident from an NNMi source node that is not configured as a managed node in HPOM, as described in step 4 on page 297 (for HPOM for Windows) and in step 4 on page 298 (for HPOM for UNIX). Verify the HPOM configuration, and then test the integration as described in step 9 of this procedure.
   - If the message indicates a problem with connecting to the HPOM server, NNMi and HPOM are not able to communicate. Continue with step 3 of this procedure.

3. Verify the accuracy and access level of the HPOM credentials by logging in to the HPOM console and displaying the HPOM active messages browser:

   - *Windows*: Log in to the computer as the **HPOM User** from the **HP NNMi–HPOM Integration Configuration** form, and then start the HPOM console.

     The user name is in the format `<Windows_domain>\<username>`.

   - *UNIX*: Log in to the HPOM console with the credentials for the **HPOM User** from the **HP NNMi–HPOM Integration Configuration** form.

   If you cannot log in to the HPOM console, contact the HPOM administrator to verify your log in credentials.

4. Verify that the connection to the HPOM management server is configured correctly:

   a. In a web browser, enter the following URL:

      <*protocol*>://<*omserver*>:<*port*>/opr-webservice//Incident.svc?wsdl

      Where the variables are related to values on the **HP NNMi–HPOM Integration Configuration** form as follows:

      — If the **HPOM SSL Enabled** check box is selected, `<protocol>` is `https`.

      — If the **HPOM SSL Enabled** check box is cleared, `<protocol>` is `http`.

— *<omserver>* is the value of **HPOM Host**.

— *<port>* is the value of **HPOM Port**.

b When prompted, enter the credentials for the **HPOM User** from the **HP NNMi–HPOM Integration Configuration** form.

The resulting web page is an XML file that describes the IWS.

— If the XML file appears, the connection to the HPOM management server is configured correctly. Continue with step 5.

— If you see an error message, the connection to the HPOM management server is not configured correctly. Contact the HPOM administrator to verify the information that you are using to connect to the HPOM web services. Continue to troubleshoot the connection to HPOM until you see the XML file.

5 Verify that the connection to NNMi is configured correctly:

If you used the information described in this step to connect to the NNMi console in step 1 of this procedure, you do not need to reconnect to the NNMi console. Continue with step 6.

a In a web browser, enter the following URL:

***<protocol>://<NNMiserver>:<port>*/nnm/**

Where the variables are related to values on the **HP NNMi–HPOM Integration Configuration** form as follows:

— If the **NNMi SSL Enabled** check box is selected, *<protocol>* is `https`.

— If the **NNMi SSL Enabled** check box is cleared, *<protocol>* is `http`.

— *<NNMiserver>* is the value of **NNMi Host**.

— *<port>* is the value of **NNMi Port**.

b When prompted, enter the credentials for an NNMi user with the Administrator role.

You should see the NNMi console. If the NNMi console does not appear, contact the NNMi administrator to verify the information that you are using to connect to NNMi. Continue to troubleshoot the connection to NNMi until the NNMi console appears.

You cannot sign into the NNMi console as a user with the Web Service Client role.

c Verify the values of the **NNMi User** and **NNMi Password**.

— If the **NNMi User** listed on the **HP NNMi–HPOM Integration Configuration** form has the Administrator role and you were able to connect to the NNMi console with this user name, then re-enter the corresponding password on the **HP NNMi–HPOM Integration Configuration** form.

— If the **NNMi User** listed on the **HP NNMi–HPOM Integration Configuration** form has the Web Service Client role, contact the NNMi administrator to verify the values of **NNMi User** and **NNMi Password**.

Passwords are hidden in the NNMi console. If you are not sure what password to specify for an NNMi user name, ask the NNMi administrator to reset the password.

6 Update the **HP NNMi–HPOM Integration Configuration** form with the values that you used for successful connections in step 4 and step 5 of this procedure.

   For more information, see HP NNMi–HPOM Integration Configuration Form Reference on page 305.

7 Click **Submit** at the bottom of the form.

8 If the status message still indicates a problem with connecting to the HPOM server, do the following:

   a Clear the web browser cache.

   b Clear all saved form or password data from the web browser.

   c Close the web browser window completely, and then re-open it.

   d Repeat step 6 and step 7 of this procedure.

9 Test the configuration by generating an incident on the NNMi management server and determining whether it reaches the HPOM management server.

HPOM Does Not Receive Some Forwarded Incidents

Verify the HPOM nodes and the incident filter.

The HPOM management server must be configured to accept incidents from the devices that NNMi manages. HPOM ignores any forwarded incident from an NNMi source node that is not configured as a managed node in HPOM, as described in step 4 on page 297 (for HPOM for Windows) and in step 4 on page 298 (for HPOM for UNIX).

If the NNMi source node is configured as a managed node in HPOM, verify the incident filter configuration on the **HP NNMi–HPOM Integration Configuration** form. Then test the filter by generating an incident on the NNMi management server and determine whether it reaches the HPOM management server.

NNMi Incident Information Is Not Available in the HPOM Messages Browser

The important information from NNMi incidents is passed to HPOM as custom message attributes. Add one or more custom messages attributes for NNMi incidents as described in step 5 on page 297 (for HPOM for Windows) and in step 5 on page 299 (for HPOM for UNIX).

NNMi and HPOM Are Not Synchronized

If either of the management servers becomes unreachable, the incidents in the NNMi incident views and the HPOM active messages browser might become mismatched. The HP NNMi–HPOM integration can re-synchronize the incidents as described here.

- If an HPOM management server becomes unavailable to the HP NNMi–HPOM integration module, the integration module periodically checks for the availability of that HPOM management server and resumes incident forwarding when a connection can be re-established. When the connection to the HPOM management server is available, the integration module forwards any incidents that might have been missed while the HPOM management server was down.

- If the NNMi management server is unavailable when an HPOM user acknowledges or unacknowledges a forwarded incident, NNMi does not receive the change of state. NNMi and HPOM might show different states for this incident.

The Integration Does Not Work Through a Firewall

Ensure that the NNMi management server is able to directly address the HPOM IWS by host and port.

HP NNMi–HPOM Integration Configuration Form Reference

The **HP NNMi–HPOM Integration Configuration** form contains the parameters for communications between NNMi and HPOM. This form is available from the **Integration Module Configuration** workspace.

Only NNMi users with the Administrator role can access the **HP NNMi–HPOM Integration Configuration** form.

The **HP NNMi–HPOM Integration Configuration** form collects information for the following general areas:

- NNMi Management Server Connection
- HPOM Management Server Connection
- Integration Behavior
- Incident Filters

To apply changes to the integration configuration, update the values on the **HP NNMi–HPOM Integration Configuration** form, and then click **Submit**.

NNMi Management Server Connection

Table 32 lists the parameters for connecting to the NNMi management server. This is the same information that you use to open the NNMi console. You can determine many of these values by examining the URL that invokes an NNMi console session. Coordinate with the NNMi administrator to determine the appropriate values for this section of the configuration form.

The default NNMi configuration uses http for connecting to the NNMi console. For information about configuring this connection to use https, see Enabling https for NNMi on page 77.

Table 32 NNMi Management Server Connection Information

Field	Description
NNMi SSL Enabled	The connection protocol specification. • If the NNMi console is configured to use https, select the **NNMi SSL Enabled** check box. • If the NNMi console is configured to use http, clear the **NNMi SSL Enabled** check box. This is the default configuration.
NNMi Host	The fully-qualified domain name of the NNMi management server. This field is pre-filled with the host name that was used to access the NNMi console. Verify 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 port that the jboss application server uses for communicating with the NNMi console, as specified in the following file: • *Windows*: `%NnmDataDir%\shared\nnm\conf\nnm.ports.properties` • *UNIX*: `$NnmDataDir/shared/nnm/conf/nnm.ports.properties` For non-SSL connections, use the value of `jboss.http.port`, 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 `jboss.https.port`, which is `443` by default.
NNMi User	The user name for connecting to the NNMi console. This user must have the NNMi Administrator or Web Service Client role.
NNMi Password	The password for the specified NNMi user.

HPOM Management Server Connection

Table 33 lists the parameters for connecting to the web services on the HPOM management server. Coordinate with the HPOM administrator to determine the appropriate values for this section of the configuration.

Table 33 HPOM Management Server Connection Information

HPOM Server Parameter	Description
HPOM SSL Enabled	The connection protocol specification. • If HPOM is configured to use https, select the **HPOM SSL Enabled** check box. This is the default configuration. • If HPOM is configured to use http, clear the **HPOM SSL Enabled** check box.
HPOM Host	The fully-qualified domain name of the HPOM management server.
HPOM Port	The port for connecting to the HPOM web services. To determine which port number to specify, do the following on the HPOM management server: • *Windows*: Examine the port settings in the IIS Manager, which is available from the **Start** menu, for example, **Start > Administrative Tools > Internet Information Services (IIS) Manager**. • *UNIX*: Run the following command: **`ovtomcatbctl -getconf`** This field is pre-filled with the value `443`, which is the default port for SSL connections to HPOM for Windows. For SSL connections to HPOM for UNIX, the default port is 8443.
HPOM User	A valid HPOM user account name with the HPOM Administrator role. This user must be permitted to view the HPOM active messages browser. *Windows only*: On the Windows operating system, HPOM works through Microsoft Internet Information Services (IIS) to authenticate user credentials. Specify a Windows user in the format *<Windows_domain>\<username>*.
HPOM Password	The password for the specified HPOM user.

Integration Behavior

Table 34 lists the parameters that describe the integration behavior. Coordinate with the NNMi administrator to determine the appropriate values for this section of the configuration.

Table 34 Integration Behavior Information

Field	Description
Forward Only	The behavior specification for the HP NNMi–HPOM integration module. By default, the integration module forwards incidents to and receives incident acknowledgements from the HPOM management servers identified on the **HP NNMi–HPOM Integration Configuration** form. You can disable the receipt of incident acknowledgements. • For one-way communication (forward incidents to HPOM but ignore incident acknowledgements from HPOM), select the **Forward Only** check box. • For two-way communication, leave the **Forward Only** check box cleared. This is the default behavior.
Holding period (minutes)	The number of minutes to wait before forwarding the configured incidents to HPOM. If an incident is closed during this time (for example, an SNMPLinkUp incident cancels an SNMPLinkDown incident), HPOM never receives that incident. If you want NNMi to forward incidents immediately, enter the value `0`. The default value is 5 minutes.
Incident Filter	A filter based on NNMi incident attributes that limits incident forwarding. The default filter (`nature=ROOTCAUSE origin=MANAGEMENTSOFTWARE`) specifies all root cause incidents that are generated by NNMi. You can modify the filter to change which incidents are forwarded to HPOM. For more information, see Incident Filters.

Incident Filters

The incident filter is the combination of all entries in the **Incident Filter** list. Filter entries with the same attribute value expand the filter (logical OR). Filter entries with different attribute values restrict the filter (logical AND). All filter entries work together; you cannot create a filter of the format `(a AND b) OR c`. Example filter entries follow the procedure.

To create the incident filter, follow these steps:

1. In the NNMi console, open the **HP NNMi–HPOM Integration Configuration** form (**Integration Module Configuration > HPOM**).
2. To delete a filter entry, in the **Incident Filter** list, edit the text to delete the entry (or entries).

   Clicking **Clear** removes all filter entries from the list.

3. To add an incident filter entry:
   a. Select an attribute from the **name** list. For the supported attributes, see the table in step c.

b Select the comparison operation to perform. Supported operators are:

- =
- !=
- <
- <=
- >
- >=

c Enter a comparison value. The following table lists the supported attributes and the acceptable values for each attribute.

Attribute	Possible Values
name	Examine the incident configuration in the NNMi console to determine the available incident names.
nature	• ROOTCAUSE • SECONDARYROOTCAUSE • SYMPTOM • SERVICEIMPACT • STREAMCORRELATION • INFO • NONE
origin	• MANAGEMENTSOFTWARE • MANUALLYCREATED • SYMPTOM • REMOTELYGENERATED • SNMPTRAP • SYSLOG • OTHER
family	• com.hp.nms.incident.family.Address • com.hp.nms.incident.family.Interface • com.hp.nms.incident.family.Node • com.hp.nms.incident.family.OSPF • com.hp.nms.incident.family.HSRP • com.hp.nms.incident.family.AggregatePort • com.hp.nms.incident.family.Board • com.hp.nms.incident.family.Connection • com.hp.nms.incident.family.Correlation

Attribute	Possible Values
category	• com.hp.nms.incident.category.Fault • com.hp.nms.incident.category.Status • com.hp.nms.incident.category.Config • com.hp.nms.incident.category.Accounting • com.hp.nms.incident.category.Performance • com.hp.nms.incident.category.Security • com.hp.nms.incident.category.Alert
severity	• NORMAL • WARNING • MINOR • MAJOR • CRITICAL

4 Repeat step 3 until all filter entries are defined.

5 Click **Submit** at the bottom of the form.

Example Incident Filters

Forward NodeDown Incidents from NNMi to HPOM

```
name=NodeDown
```

Forward NodeDown and InterfaceDown Incidents from NNMi to HPOM

```
name=NodeDown
name=InterfaceDown
```

Forward CiscoLinkDown Incidents from NNMi to HPOM

```
name=CiscoLinkDown
```

Forward NNMi Management Events with Severity of MAJOR or MINOR

```
origin=Management Software
severity=MAJOR
severity=MINOR
```

Forward NNMi Incidents with Severity of at least MINOR and nature of ROOTCAUSE or SERVICEIMPACT

```
severity>=MINOR
nature=ROOTCAUSE
nature=SERVICEIMPACT
```

Incident Filter Limitations

Because all filter entries combine to create one incident filter for the NNMi management server, the following limitations apply:

- The stated severity applies to all incidents. For example, to forward NodeDown incidents with a severity of MINOR or higher and InterfaceDown incidents with a severity of MAJOR, set the filter severity to `>=MINOR` and use HPOM logic to filter out the unwanted InterfaceDown messages.
- The incident filter does not provide a mechanism for limiting incident forwarding to specific source nodes. The HPOM managed node (or external node) configuration limits the forwarded incidents that HPOM accepts.

HP Universal Configuration Management Database

HP Universal Configuration Management Database (UCMDB) automatically maintains accurate, up-to-date information on infrastructure and application relationships through native integration to HP Discovery and Dependency mapping (DDM). UCMDB is beneficial for the following tasks:

- Using impact modeling to show the rippling effect of infrastructure and application changes before they occur.
- Tracking actual planned and unplanned changes through discovered change history.
- Gaining a shared, authoritative view of the environment through awareness of existing data repositories.

For information about purchasing UCMDB, contact your HP sales representative.

This chapter contains the following topics:

HP NNMi–HP UCMDB Integration

The HP NNMi–HP UCMDB integration shares NNMi topology information with UCMDB. UCMDB stores each device in the NNMi topology as a configuration item (CI). UCMDB applies Discovery and Dependency Mapping (DDM) patterns to the CIs for the NNMi topology to predict the impact of a device failure. This impact analysis is available from the UCMDB user interface and also from the NNMi console.

Value

The HP NNMi–HP UCMDB integration allows NNMi to be the authoritative source for network device relationships.

Supported Versions

The information in this chapter applies to the following product versions:

- UCMDB version 8.00 or higher
- NNMi version 8.10 or higher

NNMi and UCMDB cannot be installed on the same computer. The two products must be installed on different computers in either of the following configurations:

- Different operating systems. For example, the NNMi management server is a Linux system, and the UCMDB server is a Windows system.
- The same operating system. For example, the NNMi management server is a Windows system, and the UCMDB server is a second Windows system.

For the most recent information about supported hardware platforms and operating systems, refer to the support matrices for both products.

Documentation

This chapter describes how to configure NNMi to communicate with UCMDB and how to use the integration from the NNMi console.

The *HP Universal CMDB–HP Network Node Manager (NNMi) Integration Guide*, which is included on the UCMDB product media, describes how to configure UCMDB to communicate with NNMi. It also describes how to use the integration from the UCMDB user interface.

Enabling the HP NNMi–HP UCMDB Integration

1. On the UCMDB server, configure the connection between UCMDB and NNMi, and customize the integration:
   a. Deploy the UCMDB–NNMi Integration Discovery package.
   b. Set up and start the DDM probe for NNMi topology data.
   c. Set up the Network – NNM Layer2 jobs.

   For more information, see the *HP Universal CMDB–HP Network Node Manager (NNMi) Integration Guide*.
2. On the NNMi management server, configure the connection between NNMi and UCMDB:
   a. In the NNMi console, open the **HP NNMi–HP UCMDB Integration Configuration** form (**Integration Module Configuration > HP UCMDB**).
   b. Select the **Enable Integration** check box to activate the remaining fields in the form.

c Enter the information for connecting to the NNMi management server. For information about these fields, see NNMi Management Server Connection on page 318.

d Enter the information for connecting to the UCMDB server. For information about these fields, see UCMDB Server Connection on page 319.

e Click **Submit** at the bottom of the form. A new window displays a status message. If the message indicates a problem with connecting to the UCMDB server, re-open the **HP NNMi–HP UCMDB Integration Configuration** form (or press **ALT+LEFT ARROW** in the message window), and then adjust the values for connecting to the UCMDB server as suggested by the text of the error message.

3 On the NNMi management server, customize the integration:

a In the NNMi console, open the **HP NNMi–HP UCMDB Integration Configuration** form (**Integration Module Configuration > HP UCMDB**).

b Enter values for the following fields:

— **HP UCMDB Correlation Rule Prefix**

— **HP UCMDB Impact Severity Level (1–9)**

For information about these fields, see Integration Behavior on page 319.

c Click **Submit** at the bottom of the form.

Using the HP NNMi–HP UCMDB Integration

Enabling the HP NNMi–HP UCMDB integration adds two URL actions to the NNMi console:

- The **Find UCMDB Impacted CIs** action, which is described in Viewing Impacted CIs.
- The **Open CI in UCMDB** action, which is described in Viewing the UCMDB CI.

For information about using the integration from the UCMDB user interface, see the *HP Universal CMDB–HP Network Node Manager (NNMi) Integration Guide*.

Viewing Impacted CIs

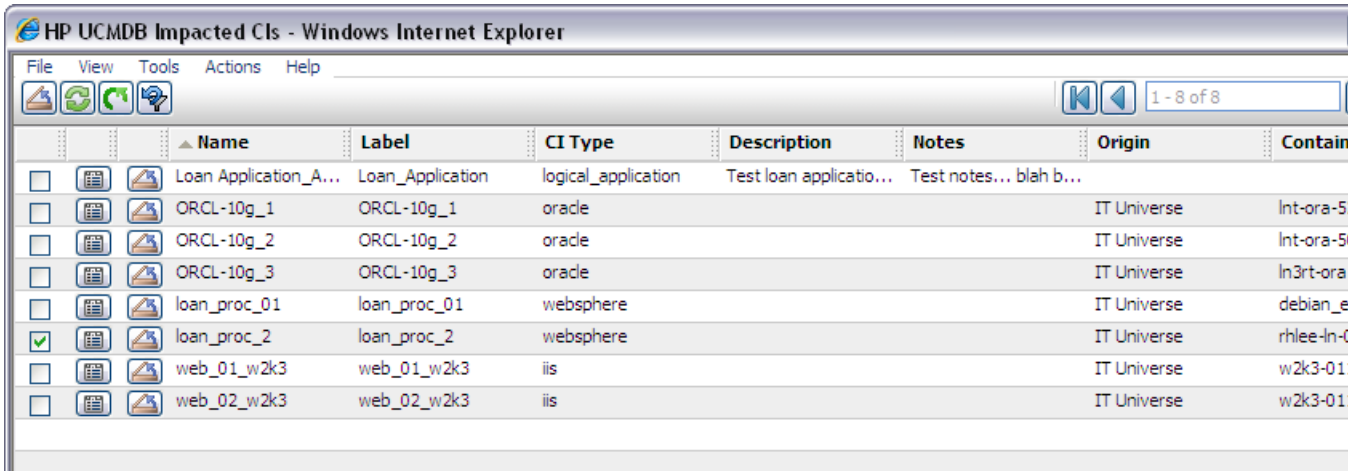
The **Find UCMDB Impacted CIs** action displays a list of the UCMDB configuration items that would be impacted (according to the correlation rules and severity level configured for the integration) for the selected node or interface. This action is available from the following NNMi console locations:

- Any node inventory view
- Any interface inventory view
- Any map view (with a node or interface selected)
- Any incident browser

The **Find UCMDB Impacted CIs** action is available for all nodes and interfaces in the NNMi topology, regardless of whether these objects are modeled in the UCMDB database.

Figure 20 shows example results from the **Find UCMDB Impacted CIs** action.

Figure 20 Example Impacted CIs Table



Viewing the UCMDB CI

The **Open CI in UCMDB** action displays the UCMDB information about the selected CI.

Changing the HP NNMi–HP UCMDB Integration Configuration

1. In the NNMi console, open the **HP NNMi–HP UCMDB Integration Configuration** form (**Integration Module Configuration > HP UCMDB**).
2. Modify the values as appropriate. For information about the fields on this form, see HP NNMi–HP UCMDB Integration Configuration Form Reference on page 317.
3. Verify that the **Enable Integration** check box at the top of the form is selected, and then click **Submit** at the bottom of the form.

The changes take effect immediately. You do not need to restart `ovjboss`.

Disabling the HP NNMi–HP UCMDB Integration

1. In the NNMi console, open the **HP NNMi–HP UCMDB Integration Configuration** form (**Integration Module Configuration > HP UCMDB**).
2. Clear the **Enable Integration** check box at the top of the form, and then click **Submit** at the bottom of the form. The integration URL actions are no longer available.

The changes take effect immediately. You do not need to restart `ovjboss`.

Troubleshooting the HP NNMi–HP UCMDB Integration

If you have verified the values in the **HP NNMi–HP UCMDB Integration Configuration** form and the status message still indicates a problem with connecting to the UCMDB server, do the following:

1. Clear the web browser cache.
2. Clear all saved form or password data from the web browser.
3. Close the web browser window completely, and then re-open it.
4. Re-enter the values in the **HP NNMi–HP UCMDB Integration Configuration** form.

Other troubleshooting information will be added as it becomes available.

For information about retrieving an updated version of this document, see Available Product Documentation on page 3.

HP NNMi–HP UCMDB Integration Configuration Form Reference

The **HP NNMi–HP UCMDB Integration Configuration** form contains the parameters for communications between NNMi and UCMDB. This form is available from the **Integration Module Configuration** workspace.

Only NNMi users with the Administrator role can access the **HP NNMi–HP UCMDB Integration Configuration** form.

The **HP NNMi–HP UCMDB Integration Configuration** form collects information for the following general areas:

- NNMi Management Server Connection
- UCMDB Server Connection
- Integration Behavior

To apply changes to the integration configuration, update the values on the **HP NNMi–HP UCMDB Integration Configuration** form, and then click **Submit**.

NNMi Management Server Connection

Table 35 lists the parameters for connecting to the NNMi management server. This is the same information that you use to open the NNMi console. You can determine many of these values by examining the URL that invokes an NNMi console session. Coordinate with the NNMi administrator to determine the appropriate values for this section of the configuration form.

The default NNMi configuration uses http for connecting to the NNMi console. For information about configuring this connection to use https, see Enabling https for NNMi on page 77.

Table 35 NNMi Management Server Information

Field	Description
HP NNMi SSL Enabled	The connection protocol specification. • If the NNMi console is configured to use https, select the **NNMi SSL Enabled** check box. • If the NNMi console is configured to use http, clear the **NNMi SSL Enabled** check box. This is the default configuration.
HP 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. Verify that this value is the name that is returned by the **`nnmofficialfqdn.ovpl -t`** command run on the NNMi management server.
HP NNMi Port	The port for connecting to the NNMi console. This field is pre-filled with the port that the jboss application server uses for communicating with the NNMi console, as specified in the following file: • *Windows*: `%NnmDataDir%\shared\nnm\conf\nnm.ports.properties` • *UNIX*: `$NnmDataDir/shared/nnm/conf/nnm.ports.properties` For non-SSL connections, use the value of `jboss.http.port`, 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 `jboss.https.port`, which is `443` by default.
HP NNMi User	The user name for connecting to the NNMi console. This user must have the NNMi Administrator or Web Service Client role.
HP NNMi Password	The password for the specified NNMi user.

UCMDB Server Connection

Table 36 lists the parameters for connecting to the web services on the UCMDB server. Coordinate with the UCMDB administrator to determine the appropriate values for this section of the configuration.

Table 36 UCMDB Management Server Information

UCMDB Server Parameter	Description
HP UCMDB SSL Enabled	The connection protocol specification for connecting to the UCMDB web services. • If the UCMDB web services are configured to use https, select the **HP UCMDB SSL Enabled** check box. This is the default configuration. • If the UCMDB web services are configured to use http, clear the **HP UCMDB SSL Enabled** check box.
HP UCMDB Host	The fully-qualified domain name of the UCMDB server.
HP UCMDB Port	The port for connecting to the UCMDB web services. If you are using the default UCMDB configuration, use port `8080` (for SSL connections to UCMDB).
HP UCMDB User	A valid UCMDB user account name with the UCMDB Administrator role.
HP UCMDB Password	The password for the specified UCMDB user.

Integration Behavior

Table 37 lists the parameters that describe the integration behavior. Coordinate with the UCMDB administrator to determine the appropriate values for this section of the configuration.

Table 37 Integration Behavior Information

Field	Description
HP UCMDB Correlation Rule Prefix	The prefix of the UCMDB impact correlation rules that apply to NNMi topology objects. The default prefix of NNM_ corresponds to the default UCMDB impact correlation rules that are provided in the `NNM_Integration.zip` file (for installation on the UCMDB management server).
HP UCMDB Impact Severity Level (1–9)	The severity level at which to apply the UCMDB impact correlation rules. HP recommends using the highest severity, 9, to include all rules in the calculation of possible impact.

nGenius Performance Manager

NetScout Systems nGenius Performance Manager provides visibility into complex networks for the following purposes:

- Application recognition and monitoring
- Analysis and troubleshooting of packets and flows
- Response time analysis
- Reporting and capacity planning
- Convergence management
- Alarming and event identification

nGenius Performance Manager leverages deep packet inspection and flow-based technologies to deliver visibility into real-time, operational intelligence that spans several types of data:

- High-level key performance indicators (KPIs), such as response time, errors, or jitter
- Application flow data, such as utilization, conversations, or top talkers
- Packet-level analysis, such as decodes and bounce diagrams

nGenius Performance Manager collects performance data from a wide variety of network data sources, allowing it to monitor the usage patterns of all network infrastructures, topologies, and applications. nGenius Performance Manager then presents the results in a collection of real-time and historical views and reports that can show the following information:

- Application performance
- Users and abusers of network resources
- Resource consumption of network capacity

For information about purchasing nGenius Performance Manager, contact your HP sales representative.

This chapter contains the following topics:

HP NNMi–nGenius Performance Manager Integration

By including nGenius Performance Manager in the HP Network Node Manager i-series Software environment, network administrators who use NNMi to monitor and manage their network devices gain application-level visibility through nGenius devices.

The HP NNMi–nGenius Performance Manager integration provides the following functionality:

- Receive nGenius Performance Manager server and probe alarms in the NNMi incident views.
- Investigate the cause of an incident by launching contextual views into nGenius Performance Manager.
- Display nGenius Probes with a NetScout icon in NNMi map views.
- Launch nGenius Performance Manager QuickViews from the NNMi console.
- Launch the nGenius Performance Manager application from the NNMi console.
- Launch NNMi Layer 2 and Layer 3 Neighbor views for the nGenius Performance Manager incidents that are available in the NNMi incident views.
- Launch the NNMi Path View map for nGenius Performance Manager incidents that are available in the NNMi incident views.
- Forward Clear Trap alarms to NNMi for each alarm generated by nGenius Performance Manager.

Value

The HP NNMi–nGenius Performance Manager integration provides the following benefits:

- Reduces the cost of delivering maximum network availability.
- Consolidates the network management infrastructure in a single console.
- Increases staff productivity and efficiency with integrated fault and application-aware performance data.
- Shrinks MTTR with contextual drill down to flow and packet-level details to identify the performance problems.

Supported Versions

The information in this chapter applies to the following product versions:

- nGenius Performance Manager version 4.3 build 450 or higher
- nGenius K2 version 1.0
- CDM Agent Firmware version 4.3 build 130 or higher
- nGenius AFMon appliance version 4.3 build 555 or higher

- nGenius InfiniStream Capture Engine version 4.5 or higher
- NNMi version 8.02 or higher

NNMi and nGenius Performance Manager Server must be installed on different computers. The NNMi management server and the nGenius Performance Manager Server computer can be of the same or different operating systems.

Documentation

The HP NNMi–nGenius Performance Manager integration is fully described in the following document:

Integrating nGenius Performance Manager v4.3 with HP Network Node Manager i Software (NetScout part number 733-0089 Rev. A, available from **http://www.netscout.com**)

Enabling the HP NNMi–nGenius Performance Manager Integration

The nGenius Performance Manager Integration utility for HP Network Node Manager installation file is available in the following location on the nGenius Performance Manager server:

- *Windows*: `%nGenius Install%\rtm\bin\nGeniusNNM8.zip`
- *UNIX*: `$nGenius Install/rtm/bin/nGeniusNNM8.zip`

As a user with administrative or root privileges, install the integration utility on the NNMi management server. The utility imports all NNMi integration-related configuration data for the nGenius Server into NNMi.

The high-level steps for installing the nGenius Performance Manager Integration utility are as follows. For detailed information, refer to *Integrating nGenius Performance Manager v4.3 with HP Network Node Manager i Software*.

1. Extract the installation files and configure NNMi support within nGenius Performance Manager.
2. Configure NetScout incidents (alarms) in NNMi.
3. Configure the nGenius Probe to send SNMP traps to port 395.
4. *Optional*. Configure Probe Router Mapping.

Using the HP NNMi–nGenius Performance Manager Integration

For information about using the HP NNMi–nGenius Performance Manager integration, refer to *Integrating nGenius Performance Manager v4.3 with HP Network Node Manager i Software*.

Disabling the HP NNMi–nGenius Performance Manager Integration

For information about disabling the HP NNMi–nGenius Performance Manager integration, refer to *Integrating nGenius Performance Manager v4.3 with HP Network Node Manager i Software*.

Troubleshooting the HP NNMi–nGenius Performance Manager Integration

For information about troubleshooting the HP NNMi–nGenius Performance Manager integration, contact NetScout Systems Customer Support. Contact information is available at:

http://www.netscout.com/support

NNM iSPI for Performance

This chapter describes considerations and best practices for integrating the NNM iSPI for Performance with NNMi.

For information about purchasing the NNM iSPI for Performance, contact your HP sales representative.

This chapter contains the following topics:

What is the NNM iSPI for Performance?

The NNM iSPI for Performance enables data collection, threshold monitoring, and threshold alarming for a predefined set of MIB variables within HP Network Node Manager i Software. From NNMi, you can launch NNM iSPI for Performance reports.

The NNMi performance monitoring settings determine the interfaces for which NNMi collects performance data and the intervals of the data collection. The NNM iSPI for Performance receives and analyzes the NNMi data and then stores the analysis results in a proprietary format.

The NNM iSPI for Performance is licensed separately from NNMi; however, both products must be licensed for the same node count.

Value

NNM iSPI for Performance adds performance reporting to NNMi for ranking, trending, and exception reporting.

- Ranking sorts elements by utilization, and highlights network elements with unusually high or unusually low utilization.
- Trending looks at a period of time and shows, through color-coding, whether performance is stable, worsening, or improving.
- Exception reporting looks at threshold conditions, and tracks threshold conditions over time.

Supported Versions

The information in this chapter applies to the following product versions:

- NNM iSPI for Performance version 8.10 or higher
- NNMi version 8.10 or higher

Documentation

The NNM iSPI for Performance release notes, support matrix, and installation instructions are available from:

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

Additionally, NNM iSPI for Performance help is installed with the product.

Enabling the NNM iSPI for Performance

The NNM iSPI for Performance can be installed as an add-on product on the NNMi management server or on a dedicated server. The recommended installation model depends on the frequency and volume of performance polling. For more information, see the NNM iSPI for Performance support matrix.

As you plan the installation, remember to add the disk space requirements and to check the RAM requirements for all software that you plan to install on a single computer.

The following steps present a high-level process for enabling the NNM iSPI for Performance:

1. On each NNMi management server, run the NNM iSPI for Performance enablement script.

   For detailed information, see the *NNM iSPI for Performance Installation Guide*.

2. Install the NNM iSPI for Performance as described in the *NNM iSPI for Performance Installation Guide*.

3. In the NNMi console, configure performance monitoring.

   For information, see *Setting Polling Characteristics in NNMi* and *Setting Thresholds for Metrics* in the NNM iSPI for Performance help.

4 In the NNM iSPI for Performance, configure performance reporting.

For information, see *Scheduling Reports* in the NNM iSPI for Performance help.

Configuring the NNM iSPI for Performance to Work with Application Failover

If you plan to configure the NNMi application failover feature, you must install the NNM iSPI for Performance on a dedicated server. In this case, the iSPI automatically connects to the new NNMi management server after failover occurs. As part of NNMi application failover configuration, the NNM iSPI for Performance enablement script must be run on each NNMi management server in the cluster.

For more information, see *Support for Application Failover* in the NNM iSPI for Performance help.

Configuring the NNM iSPI for Performance to Run Under HA

If you plan to run NNMi under high availability (HA), the following options are available:

- If the NNM iSPI for Performance is installed on the NNMi management servers, the iSPI must run in the same HA resource group as NNMi. For information about this configuration, see Configuring the NNM iSPI for Performance as an Add-on iSPI on page 140.
- If the NNM iSPI for Performance is on a dedicated server, the iSPI can run in a separate HA resource group in the NNMi HA cluster, or the iSPI can run outside of HA. In either case, the iSPI connects to the virtual hostname of the NNMi HA resource group. For information about the HA scenario, see Configuring the NNM iSPI for Performance on a Dedicated Server for HA on page 149.

As part of NNMi HA configuration, the NNM iSPI for Performance enablement script must be run on each NNMi management server in the cluster.

The NNM iSPI for Performance delivers the `PerfSPIHA` Perl module, which contains custom implementations of the HA configuration commands. When running the HA configuration commands, use the value `PerfSPIHA` in place of *<iSPI_PM_Name>* in the instructions.

Using the NNM iSPI for Performance

The NNM iSPI for Performance adds some reporting actions to the NNMi console. Each of these actions starts the NNM iSPI for Performance user interface at the designated level.

For information about the NNM iSPI for Performance reports, see the NNM iSPI for Performance help.

Backup Provisions for the NNM iSPI for Performance

The NNMi backup scripts only back up the performance monitoring configuration. NNMi does *not* provide a mechanism for backing up any performance data.

To back up the performance data, you must implement a separate backup tool on the NNM iSPI for Performance system. Add the deploy folders and all of their subfolders to this separate backup process. The NNM iSPI for Performance may be running during the backup.

The NNM iSPI for Performance stores data in the following folders:

- *Windows*:

  ```
  %NnmDataDir%\NNMPerformanceSPI\InterfaceHealth\deploy\
  %NnmDataDir%\NNMPerformanceSPI\NodeHealth\deploy\
  ```

- *UNIX*:

  ```
  $NnmDataDir/NNMPerformanceSPI/InterfaceHealth/deploy/
  $NnmDataDir/NNMPerformanceSPI/NodeHealth/deploy/
  ```

Resetting the NNM iSPI for Performance

If you want to completely reset your NNMi configuration and database so that you can start again with a clean slate, you must also reset the NNM iSPI for Performance configuration to keep the topology identifiers in agreement between the two products.

After you reset the NNMi database, run the NNM iSPI for Performance reset utility as described in *Resetting the Performance SPI* in the NNM iSPI for Performance help.

Moving the NNM iSPI for Performance to a Different Computer

You can include the NNM iSPI for Performance in your NNMi test environment. The `nnmconfigexport.ovpl` command exports the NNMi performance monitoring configuration so that you can replicate the test configuration on the production NNMi management server.

The NNM iSPI for Performance reports are based on unique identifiers for the objects in the NNMi topology. Because these identifiers will be different on the new NNMi management server, you cannot port existing reports from the test system to the production system.

For information on the scenario for moving NNMi from a test system to the production environment, see Changing the NNMi Management Server on page 189.

Disabling the NNM iSPI for Performance

For information about uninstalling the NNM iSPI for Performance, see *Uninstalling the Performance SPI* in the NNM iSPI for Performance help.

Troubleshooting the NNM iSPI for Performance

For information about uninstalling the NNM iSPI for Performance, see *Troubleshooting* in the NNM iSPI for Performance help.

Additional Information

This section contains the following appendix:

NNMi Environment Variables

HP Network Node Manager i Software provides many environment variables that are available for your use in navigating the file system and writing scripts.

This appendix contains the following topics:

Environment Variables Used in This Document

This document uses the following two NNMi environment variables to reference file and directory locations. The default values are listed here. Actual values depend upon the selections made during NNMi installation.

- Windows:
  - `%NnmInstallDir%`: `<drive>\Program Files\HP\HP BTO Software`
  - `%NnmDataDir%`: `<drive>\Documents and Settings\All Users\Application Data\HP\HP BTO Software`

On Windows systems, the NNMi installation process creates these environment variables so they are always available.

If you first installed NNMi 8.00, your system uses different values for these environment variables, as described in Windows Paths and Environment Variables from NNMi 8.00 on page 336.

- UNIX:
  - `$NnmInstallDir`: `/opt/OV`
  - `$NnmDataDir`: `/var/opt/OV`

On UNIX systems, you must manually create these environment variables if you want to use them.

Other Available Environment Variables

NNMi administrators access some NNMi file locations regularly. NNMi provides a script that sets up many environment variables for navigating to commonly accessed locations.

If you first installed NNMi 8.00, see Windows Paths and Environment Variables from NNMi 8.00 on page 336.

To set up the extended list of NNMi environment variables, use a command similar to the following examples:

- Windows: "`C:\Program Files\HP\HP BTO Software\bin\nnm.envvars.bat`"
- UNIX: `. /opt/OV/bin/nnm.envvars.sh`

After you run the command for your operating system, you can use the NNMi environment variables shown in Table 38 (Windows) or Table 39 (UNIX) to get to commonly used NNMi file locations.

Table 38 Environment Variable Locations for the Windows Operating System

Variable	Windows (example)
%NNM_BIN%	C:\Program Files\HP\HP BTO Software\bin
%NNM_CONF%	C:\Documents and Settings\All Users\Application Data\HP\HP BTO Software\conf
%NNM_DATA%	C:\Documents and Settings\All Users\Application Data\HP\HP BTO Software
%NNM_DB%	C:\Documents and Settings\All Users\Application Data\HP\HP BTO Software\databases
%NNM_JAVA%	C:\Program Files\HP\HP BTO Software\nonOV\jdk\b\bin\java.exe
%NNM_JAVA_DIR%	C:\Program Files\HP\HP BTO Software\java
%NNM_JAVA_PATH_SEP%	;
%NNM_JBOSS%	C:\Program Files\HP\HP BTO Software\nonOV\jboss\nms
%NNM_JBOSS_DEPLOY%	C:\Program Files\HP\HP BTO Software\nonOV\jboss\nms\server\nms\deploy
%NNM_JBOSS_LOG%	C:\Program Files\HP\HP BTO Software\nonOV\jboss\nms\server\nms\log
%NNM_JBOSS_ROOT%	C:\Program Files\HP\HP BTO Software\nonOV\jboss\nms
%NNM_JBOSS_SERVERCONF%	C:\Program Files\HP\HP BTO Software\nonOV\jboss\nms\server\nms
%NNM_JRE%	C:\Program Files\HP\HP BTO Software\nonOV\jdk\b
%NNM_LOG%	C:\Documents and Settings\All Users\Application Data\HP\HP BTO Software\log
%NNM_LRF%	C:\Documents and Settings\All Users\Application Data\HP\HP BTO Software\shared\nnm\lrf

Table 38 Environment Variable Locations for the Windows Operating System (cont'd)

Variable	Windows (example)
%NNM_PRIV_LOG%	C:\Documents and Settings\All Users\Application Data\HP\HP BTO Software\log
%NNM_SHARE_LOG%	C:\Documents and Settings\All Users\Application Data\HP\HP BTO Software\log
%NNM_SNMP_MIBS%	C:\Documents and Settings\All Users\Application Data\HP\HP BTO Software\share\snmp_mibs
%NNM_SUPPORT%	C:\Program Files\HP\HP BTO Software\support
%NNM_TMP%	C:\Documents and Settings\All Users\Application Data\HP\HP BTO Software\tmp
%NNM_WWW%	C:\Program Files\HP\HP BTO Software\www

Table 39 Environment Variable Locations for UNIX Operating Systems

Variable	HP-UX
$NNM_BIN	/opt/OV/bin
$NNM_CONF	/var/opt/OV/conf
$NNM_DATA	/var/opt/OV
$NNM_DB	/var/opt/OV/databases
$NNM_JAVA	/opt/OV/nonOV/jdk/b/bin/java
$NNM_JAVA_DIR	/opt/OV/java
$NNM_JAVA_PATH_SEP	:
$NNM_JBOSS	/opt/OV/nonOV/jboss/nms
$NNM_JBOSS_DEPLOY	/opt/OV/nonOV/jboss/nms/server/nms/deploy
$NNM_JBOSS_LOG	/opt/OV/nonOV/jboss/nms/server/nms/log
$NNM_JBOSS_ROOT	/opt/OV/nonOV/jboss/nms
$NNM_JBOSS_SERVERCONF	/opt/OV/nonOV/jboss/nms/server/nms
$NNM_JRE	/opt/OV/nonOV/jdk/b
$NNM_LOG	/var/opt/OV/log
$NNM_LRF	/var/opt/OV/shared/nnm/lrf
$NNM_PRIV_LOG	/var/opt/OV/log
$NNM_SHARE_LOG	/var/opt/OV/log
$NNM_SNMP_MIBS	/var/opt/OV/share/snmp_mibs

Table 39 Environment Variable Locations for UNIX Operating Systems (cont'd)

Variable	HP-UX
$NNM_SUPPORT	/opt/OV/support
$NNM_TMP	/var/opt/OV/tmp
$NNM_WWW	/opt/OV/www

Windows Paths and Environment Variables from NNMi 8.00

On the Windows operating system, the default paths to NNMi files are different for versions 8.00 and 8.01 (or higher). When you upgrade to a newer version of NNMi on top of NNMi 8.00, the newer version of NNMi continues to use the paths that were defined for NNMi 8.00. In this case, the locations of the environment variables created during NNMi installation are as follows:

- `%NnmInstallDir%`: *<drive>*`\Program Files (x86)\HP OpenView`
- `%NnmDataDir%`: *<drive>*`\Program Files (x86)\HP OpenView\data`

To set up the extended list of NNMi environment variables, use a command similar to the following example:

```
"C:\Program Files(x86)\HP OpenView\bin\nnm.envvars.bat"
```

After you run the command, you can use the NNMi environment variables shown in Table 40 to get to commonly used NNMi file locations.

For the upgrade scenario *only*, the information in Table 40 supersedes that in Table 38 on page 334.

Table 40 NNMi 8.00 Environment Variable Locations for the Windows Operating System

Variable	Windows (example)
%NNM_BIN%	C:\Program Files (x86)\HP OpenView\bin
%NNM_CONF%	C:\Program Files (x86)\HP OpenView\data\conf
%NNM_DATA%	C:\Program Files (x86)\HP OpenView\data
%NNM_DB%	C:\Program Files (x86)\HP OpenView\data\databases
%NNM_JAVA%	C:\Program Files (x86)\HP OpenView\nonOV\jdk\b\bin\java.exe
%NNM_JAVA_DIR%	C:\Program Files (x86)\HP OpenView\java
%NNM_JAVA_PATH_SEP%	;
%NNM_JBOSS%	C:\Program Files (x86)\HP OpenView\nonOV\jboss\nms
%NNM_JBOSS_DEPLOY%	C:\Program Files (x86)\HP OpenView\nonOV\jboss\nms\server\nms\deploy
%NNM_JBOSS_LOG%	C:\Program Files (x86)\HP OpenView\nonOV\jboss\nms\server\nms\log

Table 40 NNMi 8.00 Environment Variable Locations for the Windows Operating System (cont'd)

Variable	Windows (example)
%NNM_JBOSS_ROOT%	C:\Program Files (x86)\HP OpenView\nonOV\jboss\nms
%NNM_JBOSS_SERVERCONF%	C:\Program Files (x86)\HP OpenView\nonOV\jboss\nms\server\nms
%NNM_JRE%	C:\Program Files (x86)\HP OpenView\nonOV\jdk\b
%NNM_LOG%	C:\Program Files (x86)\HP OpenView\data\log
%NNM_PRIV_LOG%	C:\Program Files (x86)\HP OpenView\data\log
%NNM_SHARE_LOG%	C:\Program Files (x86)\HP OpenView\data\log
%NNM_SNMP_MIBS%	C:\Program Files (x86)\HP OpenView\data\share\snmp_mibs
%NNM_SUPPORT%	C:\Program Files (x86)\HP OpenView\support
%NNM_TMP%	C:\Program Files (x86)\HP OpenView\data\tmp
%NNM_WWW%	C:\Program Files (x86)\HP OpenView\www

Glossary

A

account

See user account.

active cluster node

See active server.

active server

The server currently running the NNMi processes in a high availability configuration.

address hint

See discovery hint.

ARP cache

The ARP (Address Resolution Protocol) cache is an operating system table that maps Data Link Layer (OSI Layer 2) addresses to Network Layer (OSI Layer 3) addresses. Data Link Layer addresses are typically MAC addresses, while Network Layer addresses are typically IP addresses. In rule-based discovery, NNMi uses ARP cache entries on discovered nodes (as well as other techniques) to find additional nodes that can be checked against the current discovery rules.

auto-discovery

See rule-based discovery.

C

Causal Engine

NNMi technology that applies root cause analysis (RCA) to network symptoms, using a causality-based approach. Causal Engine RCA is triggered by certain occurrences, including changes detected as a result of state polling, SNMP traps, and specific incidents. The Causal Engine uses RCA to determine the status of managed objects, to formulate conclusions about them, and to generate root cause incidents.

causality

Denotes the relationship between one event (the cause) and another event (the effect) which is the direct consequence (result) of the first. NNMi uses causality analysis algorithms to analyze event cycles and identify solutions for resolving network issues.

cluster

In an NNMi context, a grouping of hardware and software, linked via high availability technology or via jboss clustering capabilities, that works together to ensure functional and data continuity in the event of component overload or failure. The computers in a cluster are commonly connected to each other through high speed LANs. Clusters are usually deployed to improve availability and/or performance.

cluster member or node

In an NNMi context, a system within a high availability cluster that has been or will be configured to support NNMi high availability.

community string

A password-like mechanism used in SNMPv1 and SNMPv2C implementations to authenticate SNMP queries to SNMP agents. The community string is passed in cleartext in SNMP packets, making it vulnerable to packet sniffing. SNMPv3 provides stronger security mechanisms for authentication.

conclusion

In NNMi, supporting detail generated and used by the Causal Engine that sheds further light on how the Causal Engine determined status and root cause incidents for a managed object.

console

See NNMi console.

D

discovery hint

An IP address found by NNMi via an SNMP ARP cache query; a CDP, EDP, or other discovery protocol

query; or a ping sweep. NNMi further queries IP addresses found as discovery hints, then checks the results against the current discovery rules in rule-based discovery.

discovery process

The process by which NNMi gathers information about network nodes so that they can be placed under management. Initial discovery runs as a two-phase process, returning device inventory information and then network connectivity information.

After initial discovery, the discovery process is ongoing. In list-based discovery, this means devices in the list of seeds will be updated if their configuration changes. In rule-based discovery, new devices will also be added if they match current discovery rules. Discovery can also be initiated on demand for a device or set of devices from the NNMi console or from the command line.

See also spiral discovery, rule-based discovery, and list-based discovery.

discovery rule

A range of user-defined IP addresses and/or system object IDs (OIDs) used to limit the rule-based discovery process. Configure discovery rules in the **Discovery Configuration** portion of the NNMi console under **Auto-Discovery Rules**. See also rule-based discovery.

discovery seed

See seed.

E

embedded database

The database included with NNMi. NNMi can also be configured to use an external Oracle database instead of the embedded database for most of its tables. See also PostgreSQL.

episode

A term used in NNMi root cause analysis to refer to a specific duration, triggered by a primary failure, during which secondary failures are suppressed or are correlated under the primary failure.

F

fault polling

A key NNMi monitoring activity, in which NNMi issues ICMP pings and/or SNMP read-only queries of status MIBs for its managed interfaces, IP addresses, and SNMP agents in order to determine the state of each managed object. Users can customize the types of fault polling performed for different interface groups, node groups, and nodes under **Monitoring Configuration** in the **Configuration** workspace of the NNMi console. Fault polling is a subset of state polling.

H

HA

See high availability.

HA resource group

In modern high availability environments such as HP ServiceGuard, Veritas Cluster Server, or Microsoft Cluster Services, applications are represented as compounds of resources, such as the application itself, its shared file systems and a virtual IP address. The resources comprise an *HA resource group*, which represents an application running in a cluster environment.

high availability

Used in this guide to refer to a hardware and software configuration that provides for uninterrupted service if part of the configuration fails. High availability (HA) means that the configuration has redundant components to keep applications running at all times even if a component fails. NNMi can be configured to support one of several commercially available HA solutions.

HP Network Node Manager i Software

An HP software product (abbreviated NNMi) designed to aid network administration and to consolidate network management activities, including the ongoing discovery of network nodes, monitoring events, and network fault management. Primarily accessed via the NNMi console.

HP Network Node Manager i-series Software

Also called HP NNM i-series Software, a family of HP software products built around the HP Network Node Manager i Software product (NNMi). HP NNM i-series Software includes NNMi, NNMi Advanced, and associated iSPIs such as the NNM iSPI for Performance and the NNM iSPI for MPLS.

I

ICMP

See Internet Control Message Protocol.

incident

In NNMi, a notification of an occurrence related to your network, displayed in NNMi console incident views and forms. NNMi includes a number of **Incident Management** and **Incident Browsing** views that allow users to filter incidents based on incident attributes. Most incident views display incidents generated directly by NNMi (sometimes called *management events*). NNMi also includes views for browsing incidents generated from SNMP traps and from NNM 6.x/7.x events.

interface

A physical port used to connect a node to the network.

interface group

One of NNMi's primary filtering techniques, where interfaces are grouped together in order to apply settings to a group or filter visualizations by group. Interface groups can be used for any or all of the following: configuring monitoring, filtering table views, and customizing map views. See also node group.

Internet Control Message Protocol

One of the core protocols of the Internet protocol suite (TCP/IP). ICMP ping is used by NNMi along with SNMP queries for state polling.

iSPI

See NNM iSPI.

J

jboss application server

NNMi processes utilize a jboss application server called `ovjboss`. This application server is the main NNMi process, hosting the Java 2 Platform, Enterprise Edition (J2EE), and Enterprise Java Beans (EJB) business logic and database access capabilities. NNM iSPIs have their own jboss application servers.

JGroups

An open source technology for multicast communication and clustering; used internally by jboss for its clustering capabilities.

L

L2

See Layer 2.

L3

See Layer 3.

Layer 2

Refers to the Data Link Layer of the multi-layered communication model, Open Systems Interconnection (OSI). The data link layer moves data across the physical links in the network. NNMi Layer 2 views provide information about the physical connectivity of devices.

Layer 3

Refers to the Network Layer of the multi-layered 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. NNMi Layer 3 views provide information about connectivity from a routing perspective.

list-based discovery

A process, based on a list of seeds, that discovers and returns detailed network information *only about the nodes that you specify as seeds*. List-based discovery maintains a limited network inventory for specific queries and tasks. Contrast with rule-based discovery. See also discovery process and spiral discovery.

logical volume

A computer storage virtualization term referring to an arbitrarily sized space in a volume group that can be used as a separate file system or as a device swap space. Several of the high availability products supported by NNMi use logical volumes in their shared file systems.

M

management server

The NNMi management server is the computer system on which the NNMi software is installed. The NNMi processes and services run on the NNMi management server. (Prior NNM revisions used the term "NNM management station" for this system.)

MIB

See Management Information Base.

Management Information Base

In SNMP, the collection of data about the managed network, organized hierarchically. The data objects

within the management information base refer to characteristics of managed devices. NNMi collects network management information by making SNMP queries to and receiving SNMP traps from managed nodes using MIB data objects (sometimes called "MIB objects," "objects," or "MIBs").

N

NNM 6.x/7.x events

An NNMi term for events forwarded from older NNM management stations to NNMi. NNMi provides incident views for browsing the incidents that NNMi generates from these forwarded events.

NNM iSPI

A Smart Plug-in within the HP Network Node Manager i-series Software family. An NNM iSPI adds functionality to NNMi for a specific technology such as MPLS or for a specific domain such as network engineering.

NNMi

See HP Network Node Manager i Software.

NNMi console

The NNMi user interface. Operators and administrators use the NNMi console for network management tasks in NNMi.

node

In the network context, a computer system or device (for example, printer, router, or bridge) in a network. While nodes that are able to respond to SNMP queries provide NNMi with the most comprehensive management information, NNMi can also perform restricted management of non-SNMP nodes.

node group

One of NNMi's primary filtering techniques, where nodes are grouped together in order to apply settings to a group or filter visualizations by group. Node groups can be used for any or all of the following: configuring monitoring, filtering table views, and customizing map views. See also interface group.

O

OID

See Object Identifier.

Object Identifier

In SNMP, a numerical sequence that identifies a MIB data object. An OID consists of numbers separated by dots in which each number represents a particular data object at that level of the MIB hierarchy. The OID is the numerical equivalent of the MIB object name, for example, the MIB object name `iso.org.dod.internet.mgmt.mib-2.bgp.bgpTraps.bgpEstablished` is equivalent to its OID `1.3.6.1.2.1.15.0.1`.

ovstatus command

A command that reports the current status of the NNMi managed processes. Can be invoked from the NNMi console (**Tools** → **NNM Status**) or at a command prompt. Refer to the *ovstatus* reference page, or the UNIX manpage.

ovstart command

A command that starts the NNMi managed processes. Invoked at a command prompt. Refer to the *ovstart* reference page, or the UNIX manpage.

ovstop command

A command that stops the NNMi managed processes. Invoked at a command prompt. Refer to the *ovstop* reference page, or the UNIX manpage.

P

ping sweep

A network probe technique that sends ICMP ECHO requests to multiple IP addresses in order to determine which addresses are assigned to responsive nodes. When enabled in rule-based discovery, NNMi can use ping sweep on configured IP address ranges to find additional nodes. Some network administrators block ICMP ECHO requests because ping sweeps can be used in denial-of-service attacks.

port

In a network hardware context, a connector for passing information into and out of a network device.

PostgreSQL

An open source relational database which NNMi uses by default to store information such as topology, incidents, and configuration information. NNMi can also be configured to use Oracle instead of PostgreSQL for most of its tables.

public key certificate

Used in network security and encryption, a file that incorporates a digital signature to bind together a public key with identity information. A certificate is

used to verify that a public key belongs to an individual or organization. NNMi uses SSL certificates, which contain a public key and a private key, for authentication and encryption of client/server communication.

R

RCA

See root cause analysis.

region

In NNMi, a grouping of devices for the purpose of configuring communication settings such as timeout values and access credentials.

role

See user role.

rule

See discovery rule.

rule-based discovery

Often called *auto-discovery*, NNMi can use rule-based discovery to seek out nodes that NNMi should add to its database, following user-specified discovery rules. NNMi looks for discovery hints in data from discovered nodes, then checks these candidates against the specified discovery rules. Configure discovery rules in the **Discovery Configuration** portion of the NNMi console under **Auto-Discovery Rules**. Contrast with list-based discovery.

root cause analysis

In NNMi, root cause analysis (RCA) refers to a class of problem solving methods used by NNMi to determine root causes for network issues. In NNMi, the root cause is the actionable issue that will resolve associated problem symptoms if it is addressed. NNMi uses the identification of the root cause in two key ways: to notify the user of the actionable problem and to suppress reporting of secondary problem symptoms until the root cause issue has been resolved. Determination of root cause may result in status changes for managed objects, generation of root cause incidents, or both.

A example of how NNMi uses RCA is the scenario in which a managed router fails, and managed nodes on the other side of the router from the NNMi management server can no longer respond to state polling queries. NNMi uses RCA to determine that the state polling failures are secondary problem symptoms. It reports the router failure as the root cause incident and refrains from reporting the problem symptoms for the downstream nodes until the root cause router failure is resolved.

root cause incident

An NNMi incident in which the *Correlation Nature* attribute is set to *Root Cause*. NNMi uses root cause analysis (RCA) to establish the root cause incident as the actionable issue that will resolve associated problem symptoms if it is addressed. See root cause analysis.

S

seed

A network node that helps NNMi discover your network by acting as a starting point for the network discovery process. For example, a seed might be a core router in your management environment. Each seed is identified by an IP address or host name. Unless rule-based discovery has been configured, NNMi's discovery process is limited to list-based discovery of specified seeds.

seeded discovery

See list-based discovery.

Simple Network Management Protocol

A simple protocol operating at the application layer (Layer 7) of the OSI model, by which management information for a network element can be inspected or altered by remote users. SNMP is the predominant protocol used by NNMi to exchange network management information with agent processes on managed nodes. NNMi supports the three most common versions of SNMP: SNMPv1, SNMPv2C, and SNMPv3.

SNMP

See Simple Network Management Protocol.

SNMP trap

Network management via polling (solicited responses from SNMP agents) is an SNMP design principle that promotes simplicity. However, the protocol does provide for communication of unsolicited messages from SNMP agents to the SNMP manager process (in this case, NNMi). Unsolicited agent messages are known as "traps" and are generated by SNMP agents in response to internal state changes or fault conditions. NNMi generates incidents from received SNMP traps, displayed in the **SNMP Traps** incident browsing view.

SNMP trap storm

A high number of unsolicited SNMP agent messages that can overwhelm an SNMP manager process (in this case, NNMi). You can configure SNMP trap storm thresholds in NNMi, using the `nnmtrapconfig.ovpl` script. NNMi blocks traps when incoming trap rates exceed the specified threshold rate, until the trap rates fall below the re-arm rate.

spiral discovery

NNMi's ongoing refinement of network topology information, which includes information about inventory, containment, relationships, and connectivity in networks managed by NNMi. See also discovery process, rule-based discovery, and list-based discovery.

state

NNMi generally uses the term **state** for self-reported managed object responses related to MIB II `ifAdminStatus`, MIB II `ifOperStatus`, performance, or availability. Contrast with status.

state polling

The directed monitoring performed by NNMi's State Poller, which uses ICMP ping and SNMP queries to retrieve fault, performance, component health, and availability data from managed objects. See also fault polling.

status

In NNMi, an attribute of a managed object that indicates its overall health. The status is calculated by the Causal Engine from the managed object's outstanding conclusions. Contrast with state.

sysObjectID

See system object ID.

system object ID

In NNMi, a specialized term for an SNMP Object Identifier that identifies a model or type of network element. The system object ID is part of a network element's MIB object, which is queried by NNMi from individual nodes during discovery. Examples of network element types that can be classified by their system object IDs include any member of the HP ProCurve switch family, an HP J8715A ProCurve Switch, and an HP SNMP agent for HP IPF systems. Other vendors' network elements can be likewise classified according to their system object IDs. A key use for the system object ID is in defining NNMi Device Profiles, which specify characteristics of network elements that can be deduced once a network element's type is known.

system account

In NNMi, a special account provided for use during NNMi installation. After installation, the NNMi system account should only be used for command-line security and for recovery purposes. Contrast with user account.

T

topology (network)

In communication networks, a schematic description of the arrangement of a network, including its nodes and connections.

trap

See SNMP trap.

U

unconnected interface

From NNMi's perspective, an unconnected interface is an interface that is not connected to another device discovered by NNMi. By default, the only unconnected interfaces that NNMi monitors are those that have IP addresses *and* are contained in nodes from the **Routers** node group.

user account

In NNMi, a way to provide access to NNMi for users or groups of users. NNMi user accounts are set up in the NNMi console and implement predetermined user roles. See system account and user role.

user role

As part of setting up user access, the NNMi administrator assigns a pre-configured user role to each NNMi user account. User roles determine which user accounts have access to the NNMi console, as well as which workspaces and actions are available to each user account. NNMi provides the following hierarchical user roles, which are predefined by the program and cannot be modified: *Administrator*, *Web Service Client*, *Operator Level 2*, *Operator Level 1*, *Guest*. See also user account.

V

virtual host name

The host name associated with a virtual IP address.

virtual IP address

An IP address that is not tied to any particular network hardware, used in high availability configurations to send uninterrupted network traffic to the most appropriate server based on current failover or load-balancing needs.

volume group

A computer storage virtualization term referring to one or more disk drives that are configured to form a single large storage area. Several of the high availability products supported by NNMi use volume groups in their shared file systems.

Index

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X


invent


Java™
COMPATIBLE