HP Network Node Manager i Software

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

Software Version: 9.0x Patch 2 (9.01)

Upgrade Reference

Document Release Date: September 2010
Software Release Date: September 2010

Legal Notices

Warranty

Restricted Rights Legend

Copyright Notices

Trademark Notices

Oracle Technology — Notice of Restricted Rights

Acknowledgements

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

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

This product includes software developed by The Legion Of The Bouncy Castle. (http://www.bouncycastle.org)

This product contains software developed by Trantor Standard Systems Inc.. (http://www.trantor.ca)

September 2010

Available Product Documentation

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

- *HP Network Node Manager i Software Documentation List*—Available on the HP manuals web site. Use this file to track additions to and revisions within the NNMi documentation set for this version of NNMi. Click a link to access a document on the HP manuals web site.
- *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 Deployment Reference*—Available on the HP manuals web site.
- *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 iSPI Network Engineering Toolset Planning and Installation Guide*—Available on the NNM iSPI NET diagnostics server product media.

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/go/hpsoftwaresupport

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 register for an HP Passport ID, go to:

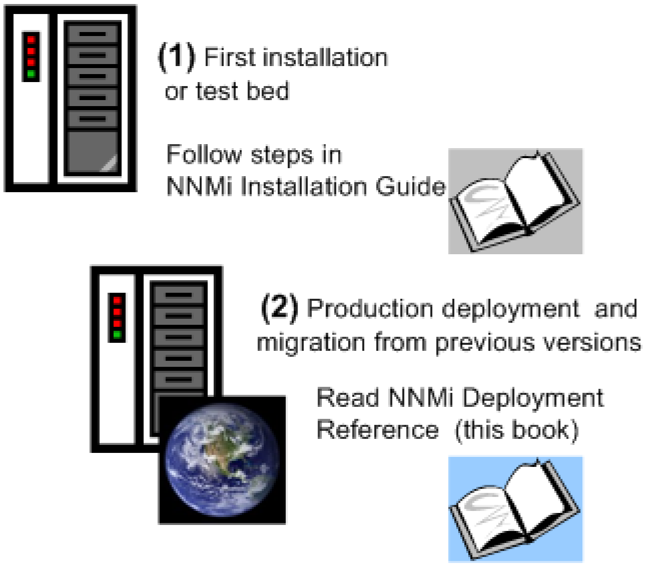
http://h20229.www2.hp.com/passport-registration.html

To find more information about access levels, go to:

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

Contents

About This Guide


(1) First installation or test bed
Follow steps in NNMi Installation Guide
(2) Production deployment and migration from previous versions
Read NNMi Deployment Reference (this book)

This chapter contains the following topics:

- What Is in This Guide?
- Environment Variables Used in This Document
- Revision History

What Is in This Guide?

This guide contains information for upgrading from HP Network Node Manager version 6.x or 7.x to HP Network Node Manager i Software version 9.0x. This information is current for the product and patch version indicated in the footer of this document.

This guide is for an expert system administrator, network engineer, or HP support engineer with experience deploying and managing networks in large installations.

The information in this guide was formerly published in the *NNMi Deployment Reference*.

HP updates this guide between product releases, 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 primarily uses the following two NNMi environment variables to reference file and directory locations. This list shows the default values. Actual values depend on the selections that you made during NNMi installation.

- *Windows Server 2008*:
  - `%NnmInstallDir%`: *<drive>*`\Program Files\HP\HP BTO Software`
  - `%NnmDataDir%`: *<drive>*`\ProgramData\HP\HP BTO Software`
- *Windows Server 2003*:
  - `%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 system environment variables, so they are always available to all users.

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

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

Additionally, this document references some of the NNMi environment variables that you can source as part of your user log-on configuration on the NNMi management server. These variables are of the form `NNM_*`. For information about this extended list of NNMi environment variables, see “Other Available Environment Variables” in the *NNMi Deployment Reference*.

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 9.0x:	
September 2010 (patch 2/ 9.01)	First English edition. Separated content for upgrading from NNM 6.x/7.x to NNMi 9.0x from the *NNMi Deployment Reference*. Minor content changes only.

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. Refer to this chapter as you plan and configure NNMi.

This chapter contains the following topics:

- Network Discovery
- Status Monitoring
- Customizing Event Monitoring

Network Discovery

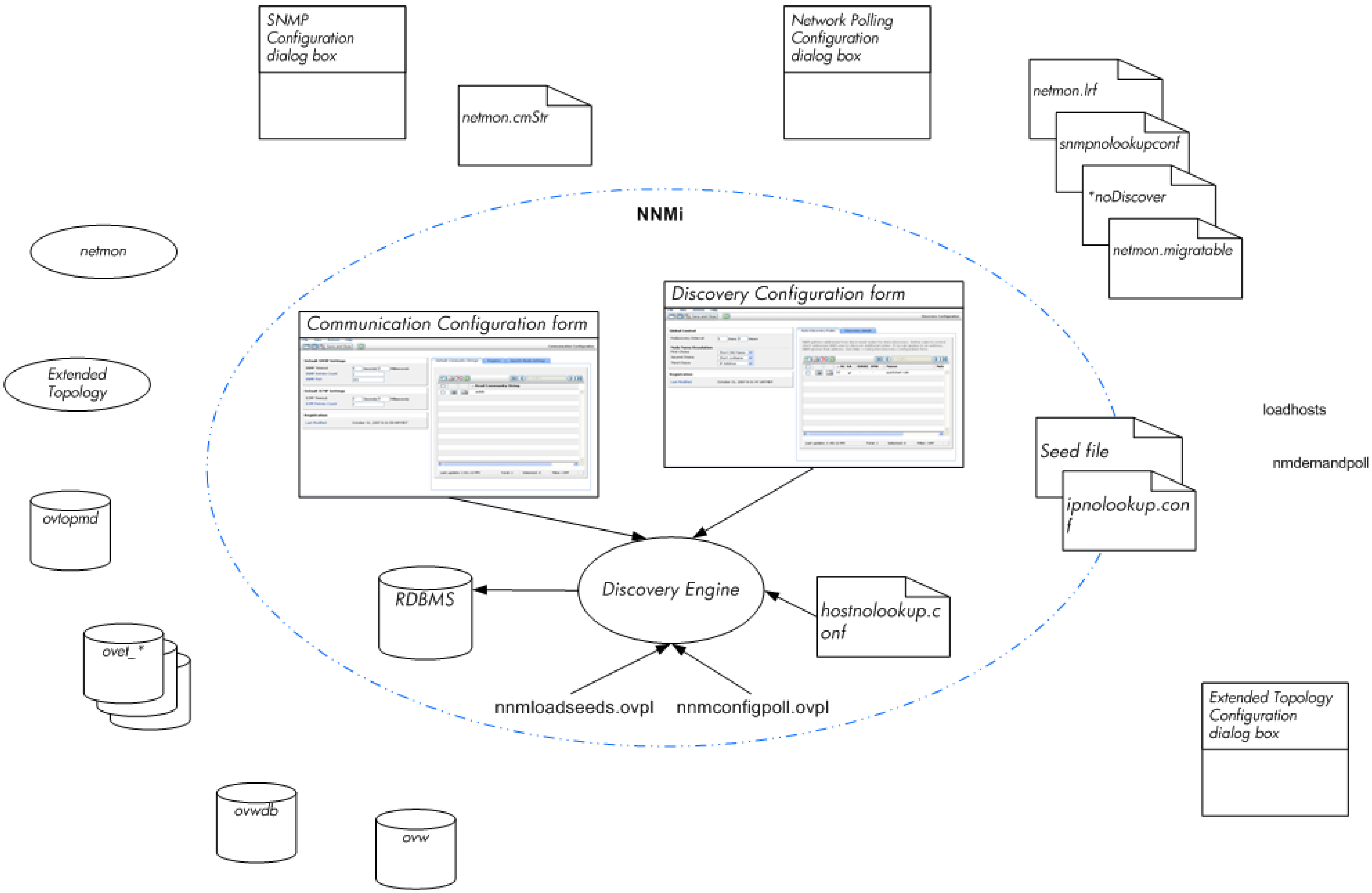
Discovery controls the network elements (devices, nodes, and their components) that are added into the database. In NNMi, “inventory discovery” refers to the activity of finding new nodes and “Layer 2 discovery” refers to the connectivity modeling previously performed by Extended Topology discovery.

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 ranges and specify at least one seed device (usually a router) before any discovery takes place.

The center of Figure 1 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 1 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, see *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.

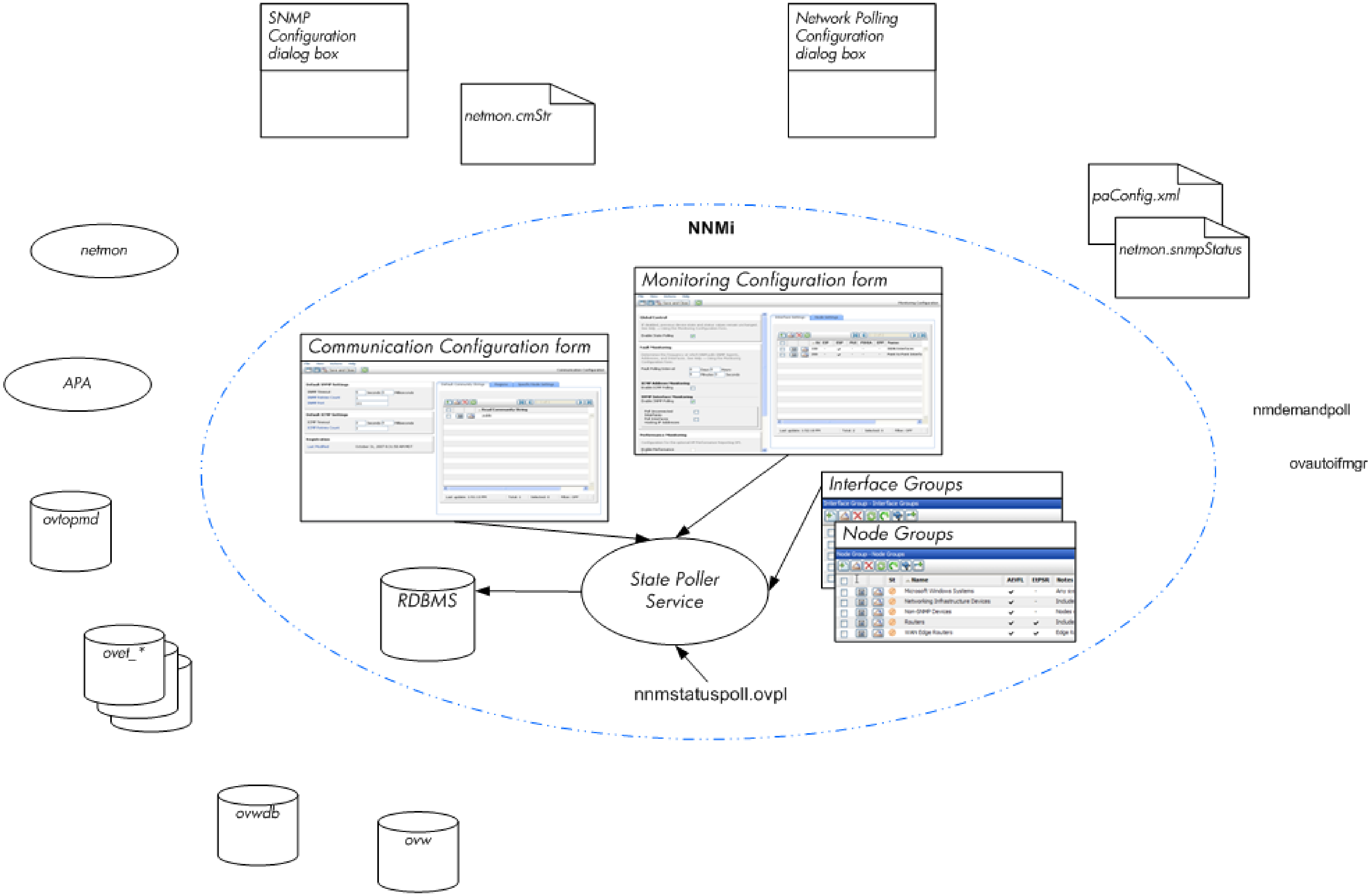
Status Monitoring

Status monitoring ensures that your network visualization is up-to-date in terms of devices or components that might 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 2 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 2 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, see *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.

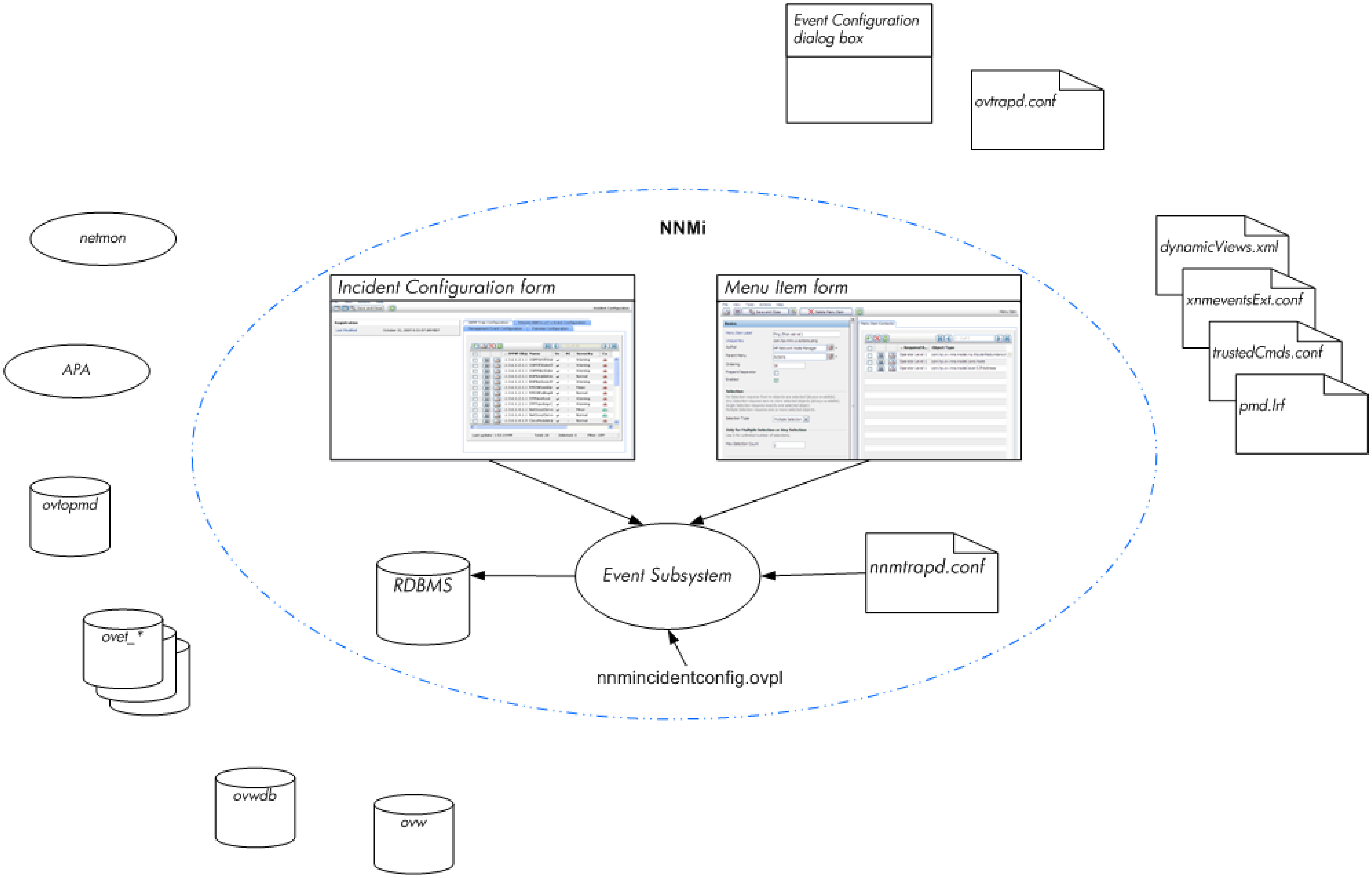
Customizing Event Monitoring

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 3 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 3 Event Monitoring Configuration Elements

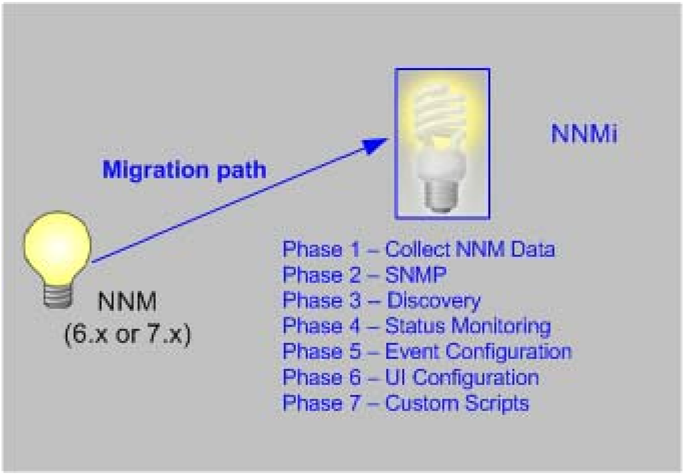


Key Concepts for Event Monitoring

This section briefly describes the main areas of change from NNM 6.x/7.x to NNMi. For more information about NNMi incidents, see *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


Migration path
NNMi
NNM
(6.x or 7.x)
Phase 1 – Collect NNM Data
Phase 2 – SNMP
Phase 3 – Discovery
Phase 4 – Status Monitoring
Phase 5 – Event Configuration
Phase 6 – UI Configuration
Phase 7 – Custom Scripts

This chapter provides a basic path for upgrading 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 basic path should serve the needs of most users. This chapter does not cover advanced upgrade 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 and all 9.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 tools that help with the upgrade process 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 Reference*. You do not need to read this chapter.

Upgrading in Phases

For some organizations, a phased approach to upgrading 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 upgrading 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 on page 61.

Table 1 presents a high-level overview of the upgrade process for the two ends of the upgrade 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 replicating an NNM configuration in NNMi. The text in the left margin indicates how the specific steps fit into the upgrade 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 do on the NNMi management server. You can perform enhancements during the upgrade 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 1 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 upgrade 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

NNMi provides tools that run on the NNM management station to collect the majority of data needed for replicating 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.

For information about the data collection tools and the information that these tools collect, see Data Collection Tools on page 56.

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 52.

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` might 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

Configure 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, replicate that information in NNMi.

Customize NNMi device profiles for the custom devices in your network.

Configure SNMP Access

NNMi discovery requires SNMP access to the managed nodes 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 24 (for the upgrade tool approach) or step 1 on page 23 (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 the upgrade process.

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:

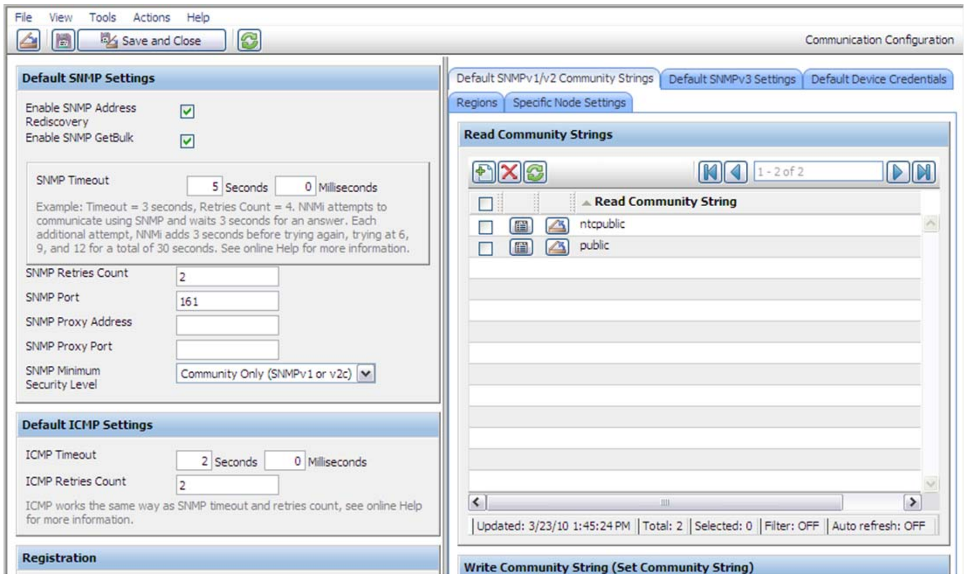
```
public
mycommstr
```

— *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 the unique values on the **Default Community Strings** tab.

c Configure timeout, retries, and port.



Default SNMP Settings

Enable SNMP Address Rediscovery ☑

Enable SNMP GetBulk ☑

SNMP Timeout 5 Seconds 0 Milliseconds

Example: Timeout = 3 seconds, Retries Count = 4. NNMi attempts to communicate using SNMP and waits 3 seconds for an answer. Each additional attempt, NNMi adds 3 seconds before trying again, trying at 6, 9, and 12 for a total of 30 seconds. See online Help for more information.

SNMP Retries Count 2

SNMP Port 161

SNMP Proxy Address

SNMP Proxy Port

SNMP Minimum Security Level Community Only (SNMPv1 or v2c)

Default ICMP Settings

ICMP Timeout 2 Seconds 0 Milliseconds

ICMP Retries Count 2

ICMP works the same way as SNMP timeout and retries count, see online Help for more information.

Registration

Read Community Strings

☐			Read Community String
☐			ntcpublic
☐			public

Updated: 3/23/10 1:45:24 PM | Total: 2 | Selected: 0 | Filter: OFF | Auto refresh: OFF

Write Community String (Set Community String)

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 28.

File name capitalization differs between NNM and NNMi. NNM uses the file name `ipNoLookup.conf`, while NNMi uses the file name `ipnolookup.conf`. NNMi does not correctly interpret anything other than all lowercase characters for this file name.

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 26) 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 about the format of these configuration files, see 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, select **Discovery Configuration** from the **Configuration** workspace.
2. 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`). NNMi maps attributes to a device according to its **device profile**, which is based on the `sysObjectID`. Device profiles group nodes for monitoring, filtering views, and categorizing 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` file 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

Configure 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, replicate any connections between devices that were configured manually in NNM.

Schedule Discovery

The NNM discovery processes can run independently. To upgrade 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.

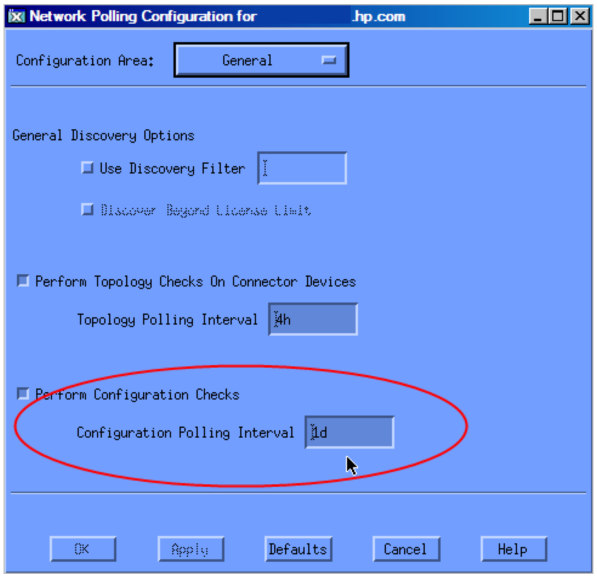
  a In a user interface, select **Options** > **Network Polling Configuration**.

  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.


Network Polling Configuration for .hp.com

Configuration Area: General

General Discovery Options

□ Use Discovery Filter

□ Discover Beyond License Limit

□ Perform Topology Checks On Connector Devices

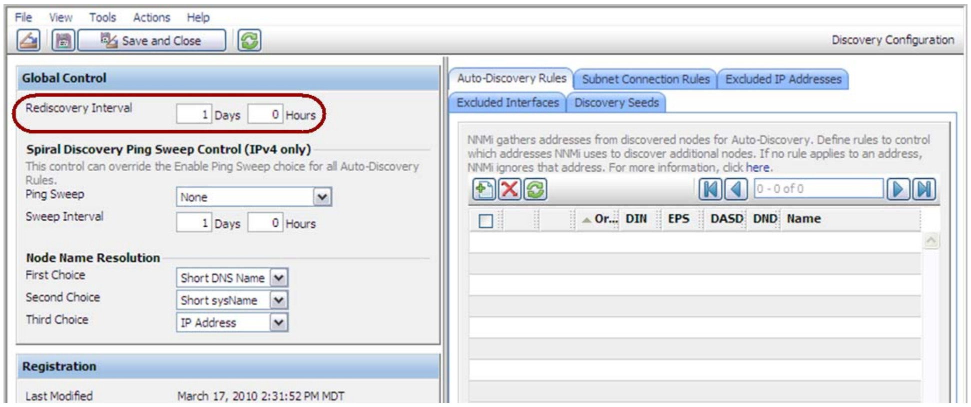
Topology Polling Interval 4h

□ Perform Configuration Checks

Configuration Polling Interval 1d

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.



Global Control

Rediscovery Interval 1 Days 0 Hours

Spiral Discovery Ping Sweep Control (IPv4 only)

This control can override the Enable Ping Sweep choice for all Auto-Discovery Rules.

Ping Sweep: None

Sweep Interval 1 Days 0 Hours

Node Name Resolution

First Choice: Short DNS Name

Second Choice: Short sysName

Third Choice: IP Address

Registration

Last Modified March 17, 2010 2:31:52 PM MDT

NNMi gathers addresses from discovered nodes for Auto-Discovery. Define rules to control which addresses NNMi uses to discover additional nodes. If no rule applies to an address, NNMi ignores that address. For more information, click here.

□			Or...	DIN	EPS	DASD	DND	Name

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 37
- Automatic discovery based on seeds and auto-discovery rules. Complete both of the following tasks:
  - — Configure Auto-Discovery Rules on page 32
  - — Add Seeds to NNMi for Seeded Discovery on page 37

For more information about the differences between the NNMi discovery methods, see *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” in the *NNMi Deployment Reference*.

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 (such as aggregated ports and vlans)
- 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, see 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, see *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 37.

   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 }
```

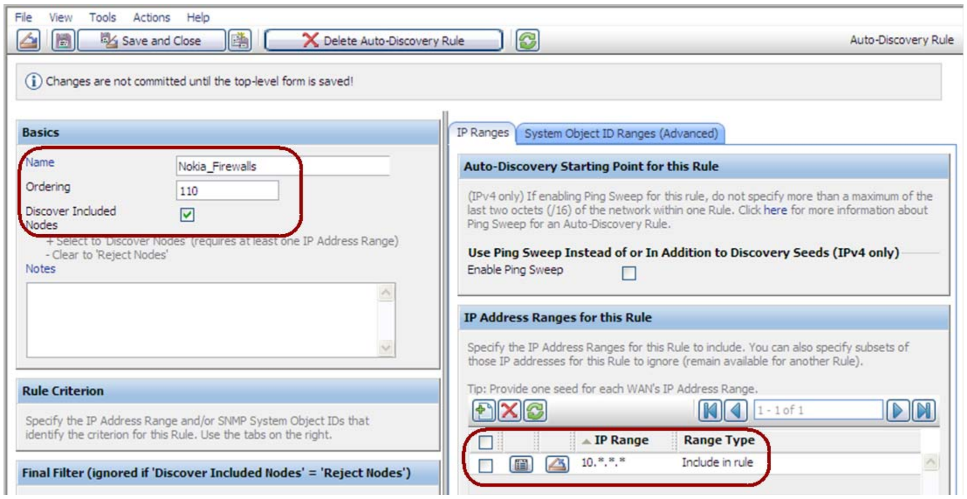
Replicate in NNMi

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 34 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.*.*.*.



ⓘ Changes are not committed until the top-level form is saved!

Basics

Name	Nokia_Firewalls
Ordering	110
Discover Included Nodes	☑

+ Select to 'Discover Nodes' (requires at least one IP Address Range)
- Clear to 'Reject Nodes'

Notes

Rule Criterion

Specify the IP Address Range and/or SNMP System Object IDs that identify the criterion for this Rule. Use the tabs on the right.

Final Filter (ignored if 'Discover Included Nodes' = 'Reject Nodes')

Auto-Discovery Starting Point for this Rule

(IPv4 only) If enabling Ping Sweep for this rule, do not specify more than a maximum of the last two octets (/16) of the network within one Rule. Click here for more information about Ping Sweep for an Auto-Discovery Rule.

Use Ping Sweep Instead of or In Addition to Discovery Seeds (IPv4 only)

Enable Ping Sweep ☐

IP Address Ranges for this Rule

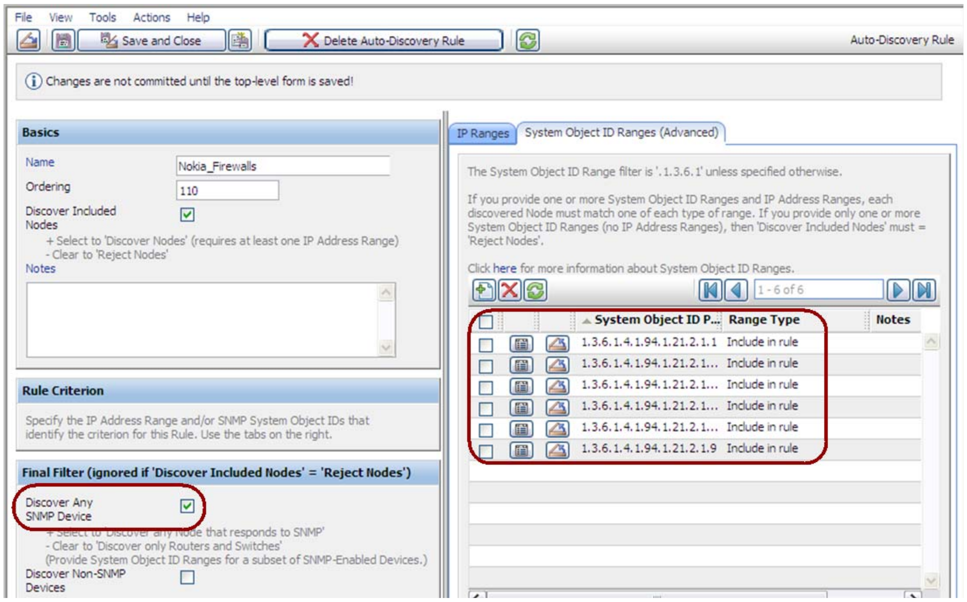
Specify the IP Address Ranges for this Rule to include. You can also specify subsets of those IP addresses for this Rule to ignore (remain available for another Rule).

Tip: Provide one seed for each WAN's IP Address Range.

1 - 1 of 1

☐			IP Range	Range Type
☐			10.*.*.*	Include in rule

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`.)



Changes are not committed until the top-level form is saved!

Basics

Name: Nokia_Firewalls

Ordering: 110

Discover Included Nodes ☑
+ Select to 'Discover Nodes' (requires at least one IP Address Range)
- Clear to 'Reject Nodes'

Notes

Rule Criterion

Specify the IP Address Range and/or SNMP System Object IDs that identify the criterion for this Rule. Use the tabs on the right.

Final Filter (ignored if 'Discover Included Nodes' = 'Reject Nodes')

Discover Any SNMP Device ☑
+ Select to 'Discover any Node that responds to SNMP'
- Clear to 'Discover only Routers and Switches'
(Provide System Object ID Ranges for a subset of SNMP-Enabled Devices.)

Discover Non-SNMP Devices □

IP Ranges | System Object ID Ranges (Advanced)

The System Object ID Range filter is '.1.3.6.1' unless specified otherwise.

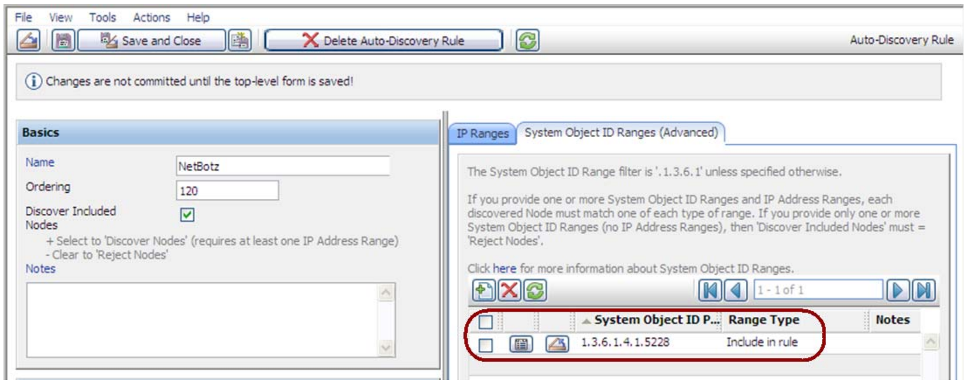
If you provide one or more System Object ID Ranges and IP Address Ranges, each discovered Node must match one of each type of range. If you provide only one or more System Object ID Ranges (no IP Address Ranges), then 'Discover Included Nodes' must = 'Reject Nodes'.

Click here for more information about System Object ID Ranges.

1 - 6 of 6

System Object ID P...	Range Type	Notes
1.3.6.1.4.1.94.1.21.2.1.1	Include in rule	
1.3.6.1.4.1.94.1.21.2.1...	Include in rule	
1.3.6.1.4.1.94.1.21.2.1...	Include in rule	
1.3.6.1.4.1.94.1.21.2.1...	Include in rule	
1.3.6.1.4.1.94.1.21.2.1...	Include in rule	
1.3.6.1.4.1.94.1.21.2.1.9	Include in rule	

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.



Changes are not committed until the top-level form is saved!

Basics

Name: NetBotz

Ordering: 120

Discover Included Nodes ☑
+ Select to 'Discover Nodes' (requires at least one IP Address Range)
- Clear to 'Reject Nodes'

Notes

IP Ranges | System Object ID Ranges (Advanced)

The System Object ID Range filter is '.1.3.6.1' unless specified otherwise.

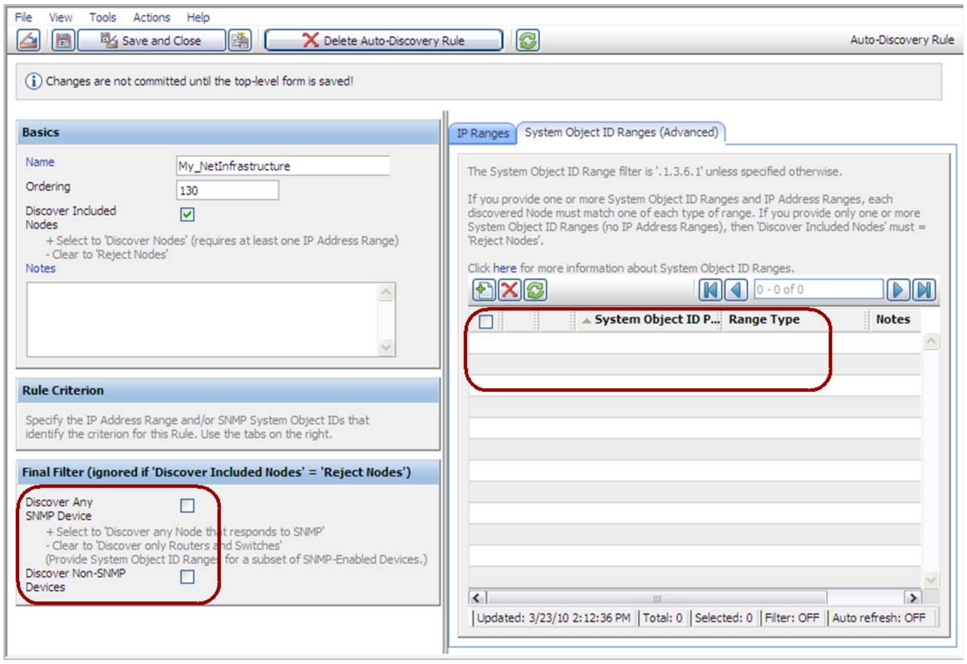
If you provide one or more System Object ID Ranges and IP Address Ranges, each discovered Node must match one of each type of range. If you provide only one or more System Object ID Ranges (no IP Address Ranges), then 'Discover Included Nodes' must = 'Reject Nodes'.

Click here for more information about System Object ID Ranges.

1 - 1 of 1

System Object ID P...	Range Type	Notes
1.3.6.1.4.1.5228	Include in rule	

d. The final rule is for switches and routers. Because NNMi discovers these devices by default, do not specify system object IDs. Only specify the IP address range.



Changes are not committed until the top-level form is saved!

Basics

Name: My_NetInfrastructure

Ordering: 130

Discover Included Nodes: ☑

+ Select to 'Discover Nodes' (requires at least one IP Address Range)
- Clear to 'Reject Nodes'

Notes

Rule Criterion

Specify the IP Address Range and/or SNMP System Object IDs that identify the criterion for this Rule. Use the tabs on the right.

Final Filter (ignored if 'Discover Included Nodes' = 'Reject Nodes')

Discover Any SNMP Device: □

+ Select to 'Discover any Node that responds to SNMP'
- Clear to 'Discover only Routers and Switches'
(Provide System Object ID Ranges for a subset of SNMP-Enabled Devices.)

Discover Non-SNMP Devices: □

The System Object ID Range filter is '.1.3.6.1' unless specified otherwise.

If you provide one or more System Object ID Ranges and IP Address Ranges, each discovered Node must match one of each type of range. If you provide only one or more System Object ID Ranges (no IP Address Ranges), then 'Discover Included Nodes' must = 'Reject Nodes'.

Click here for more information about System Object ID Ranges.

0 - 0 of 0

□	System Object ID P...	Range Type	Notes

Updated: 3/23/10 2:12:36 PM | Total: 0 | Selected: 0 | Filter: OFF | Auto refresh: OFF

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, see 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 about how to determine discovery status, see *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 add
      ```

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

   b. Edit the template file (`add.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 38:

```
<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 add.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>
```

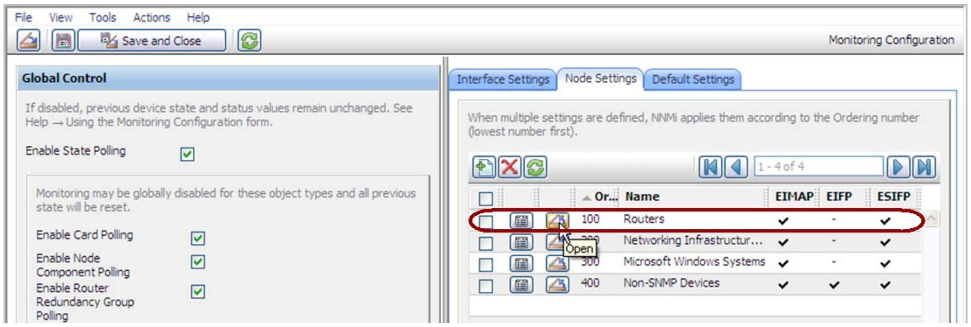
Replicate to NNMi

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.



If disabled, previous device state and status values remain unchanged. See Help → Using the Monitoring Configuration form.

Enable State Polling ☑

Monitoring may be globally disabled for these object types and all previous state will be reset.

Enable Card Polling ☑

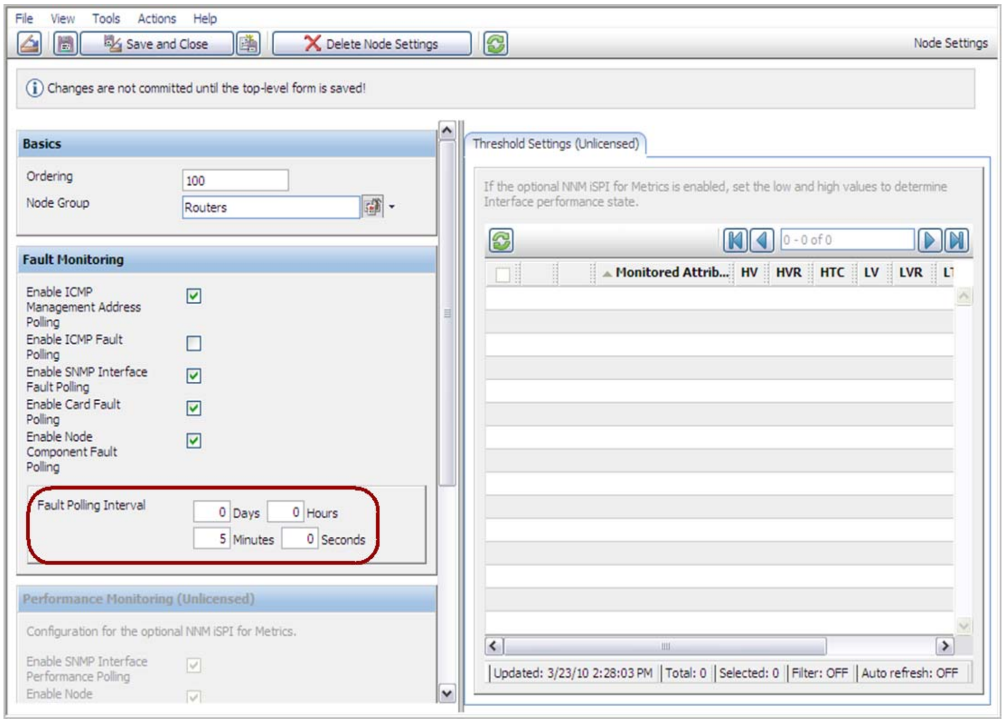
Enable Node Component Polling ☑

Enable Router Redundancy Group Polling ☑

When multiple settings are defined, NNMi applies them according to the Ordering number (lowest number first).

Or...	Name	EIMAP	EIFP	ESIFP
100	Routers	✓	-	✓
200	Networking Infrastructur...	✓	-	✓
300	Microsoft Windows Systems	✓	-	✓
400	Non-SNMP Devices	✓	✓	✓

3 Set the **Fault Polling Interval** for the group.



Changes are not committed until the top-level form is saved!

Ordering 100

Node Group Routers

Enable ICMP Management Address Polling ☑

Enable ICMP Fault Polling ☐

Enable SNMP Interface Fault Polling ☑

Enable Card Fault Polling ☑

Enable Node Component Fault Polling ☑

Fault Polling Interval 0 Days 0 Hours 5 Minutes 0 Seconds

Configuration for the optional NNM iSPI for Metrics.

Enable SNMP Interface Performance Polling ☑

Enable Node ☑

If the optional NNM iSPI for Metrics is enabled, set the low and high values to determine Interface performance state.

Monitored Attrib...	HV	HVR	HTC	LV	LVR	LT

Updated: 3/23/10 2:28:03 PM | Total: 0 | Selected: 0 | Filter: OFF | Auto refresh: OFF

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

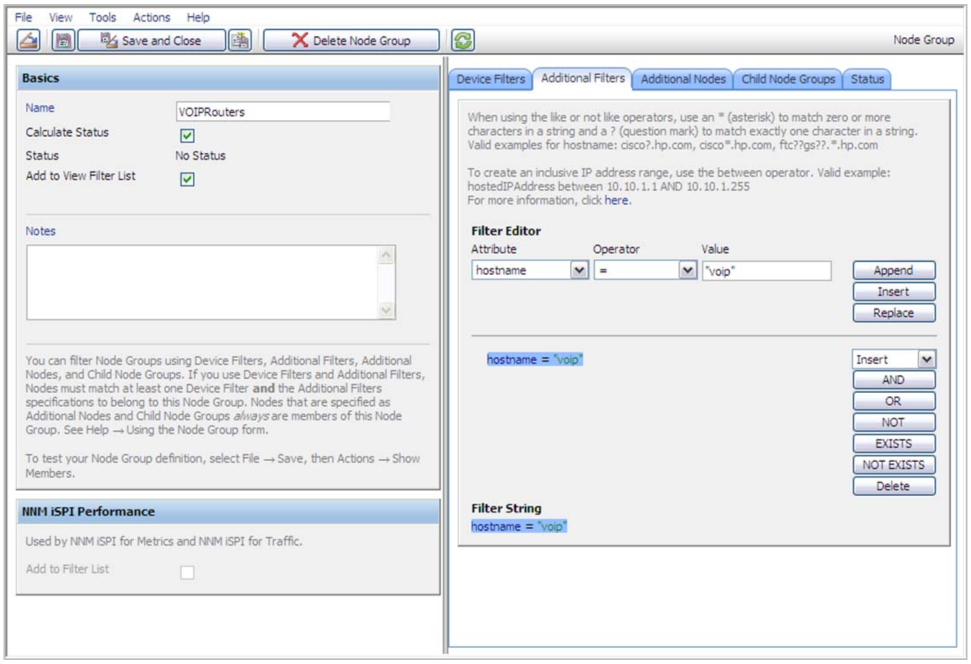
NNMi polling process

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. Save and close this form.



Basics

Name: VOIPRouters

Calculate Status ☑

Status: No Status

Add to View Filter List ☑

Notes

You can filter Node Groups using Device Filters, Additional Filters, Additional Nodes, and Child Node Groups. If you use Device Filters and Additional Filters, Nodes must match at least one Device Filter **and** the Additional Filters specifications to belong to this Node Group. Nodes that are specified as Additional Nodes and Child Node Groups *always* are members of this Node Group. See Help → Using the Node Group form.

To test your Node Group definition, select File → Save, then Actions → Show Members.

NNM iSPI Performance

Used by NNM iSPI for Metrics and NNM iSPI for Traffic.

Add to Filter List □

When using the like or not like operators, use an * (asterisk) to match zero or more characters in a string and a ? (question mark) to match exactly one character in a string. Valid examples for hostname: cisco?.hp.com, cisco*.hp.com, ftc??gs??.*.hp.com

To create an inclusive IP address range, use the between operator. Valid example: hostedIPAddress between 10.10.1.1 AND 10.10.1.255
For more information, click here.

Filter Editor

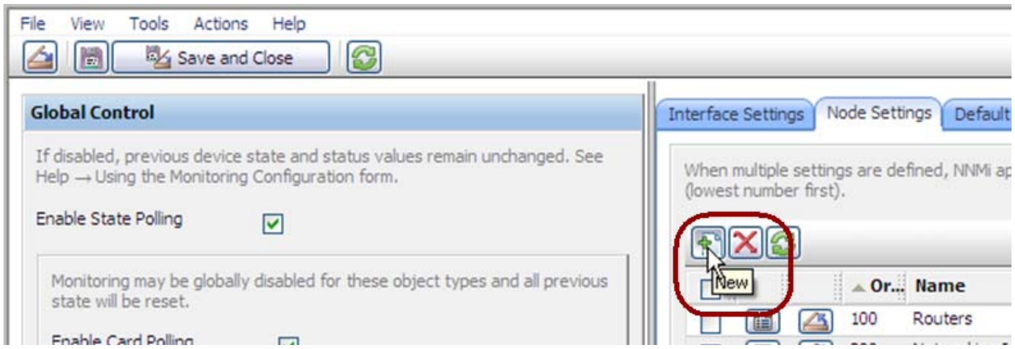
Attribute	Operator	Value
hostname	=	"voip"

hostname = "voip"

Filter String

hostname = "voip"

2 On the **Monitoring Configuration** form, add new node settings, as shown here.



Global Control

If disabled, previous device state and status values remain unchanged. See Help → Using the Monitoring Configuration form.

Enable State Polling ☑

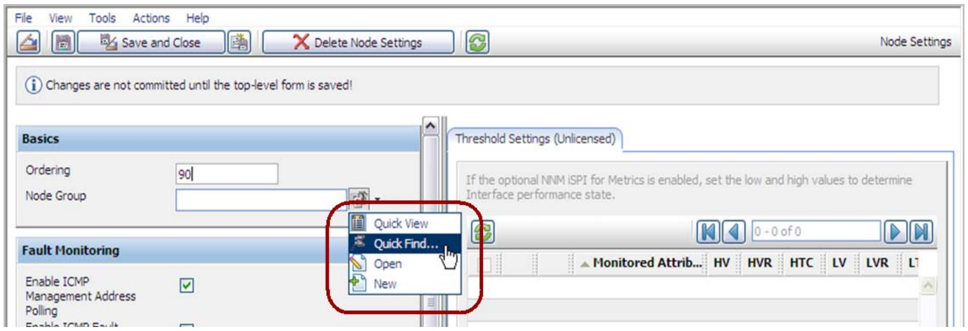
Monitoring may be globally disabled for these object types and all previous state will be reset.

Enable Card Polling

When multiple settings are defined, NNMi ap (lowest number first).

Or...	Name
100	Routers

3 Specify an ordering value, and then select quick find for the **Node Group** field, as shown here.



Changes are not committed until the top-level form is saved!

Basics

Ordering: 90

Node Group:

Fault Monitoring

Enable ICMP Management Address Polling ☑

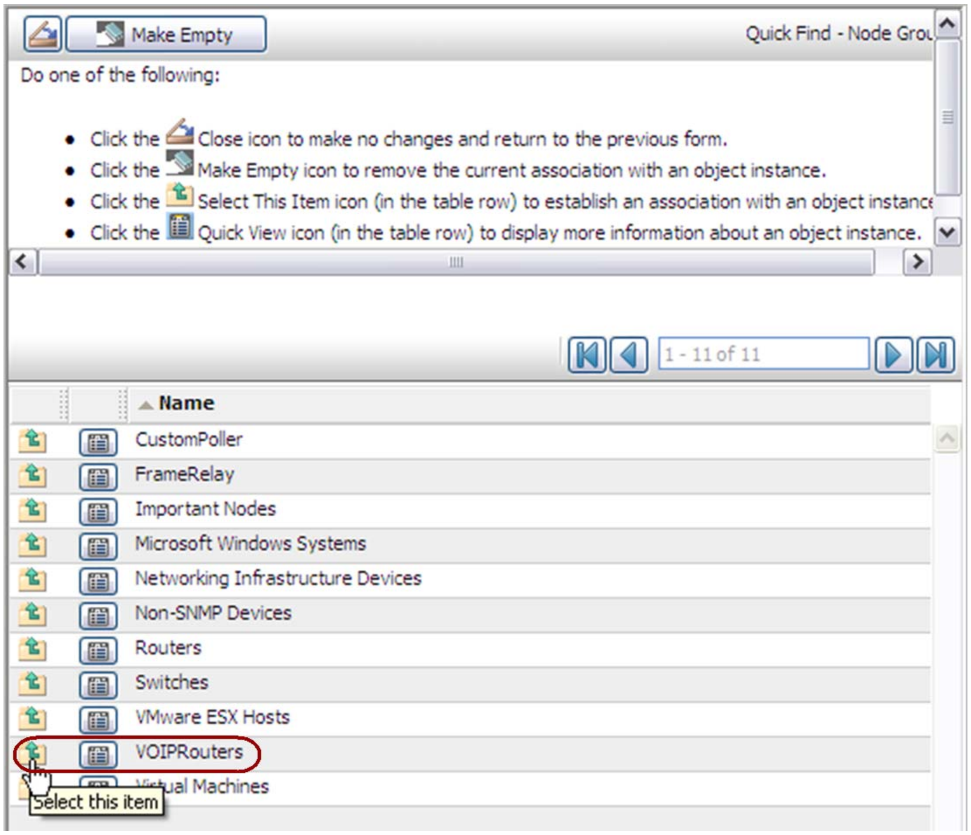
Threshold Settings (Unlicensed)

If the optional NNM iSPI for Metrics is enabled, set the low and high values to determine Interface performance state.

0 - 0 of 0

Monitored Attrib...	HV	HVR	HTC	LV	LVR	LT

4 Select the VOIPRouters node group, as shown here.



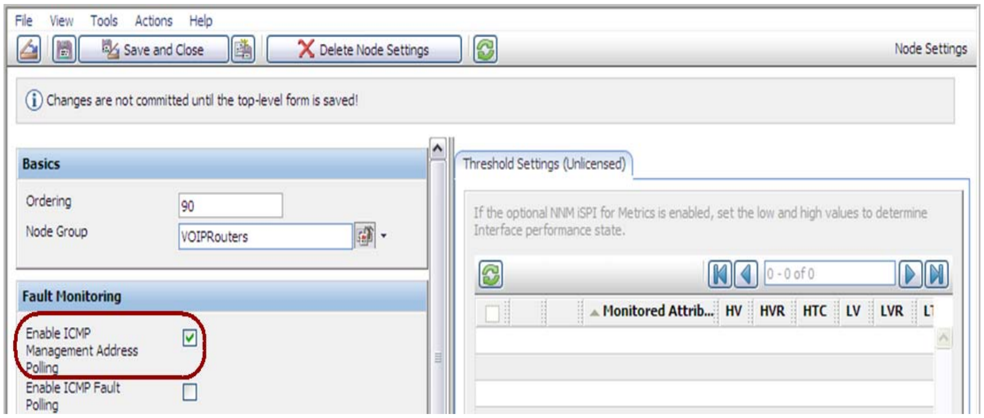
Do one of the following:

- Click the Close icon to make no changes and return to the previous form.
- Click the Make Empty icon to remove the current association with an object instance.
- Click the Select This Item icon (in the table row) to establish an association with an object instance
- Click the Quick View icon (in the table row) to display more information about an object instance.

1 - 11 of 11

Name
CustomPoller
FrameRelay
Important Nodes
Microsoft Windows Systems
Networking Infrastructure Devices
Non-SNMP Devices
Routers
Switches
VMware ESX Hosts
VOIPRouters
Virtual Machines

5 Verify that the **Enable ICMP Management Address Polling** check box is selected, as shown here. Save and close the form.



Changes are not committed until the top-level form is saved!

Ordering: 90

Node Group: VOIPRouters

Enable ICMP Management Address Polling ☑

Enable ICMP Fault Polling ☐

If the optional NNM iSPI for Metrics is enabled, set the low and high values to determine Interface performance state.

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

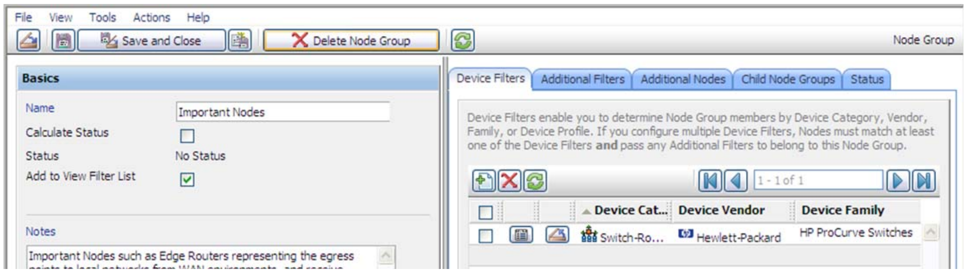
NNMi polling process

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.



Basics

Name	Important Nodes
Calculate Status	☐
Status	No Status
Add to View Filter List	☑

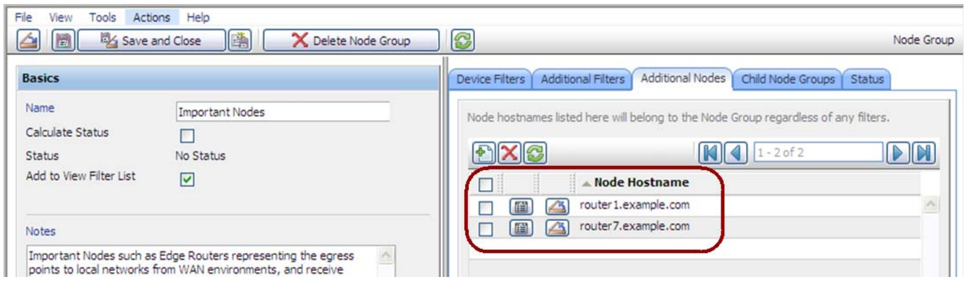
Notes

Important Nodes such as Edge Routers representing the egress

Device Filters enable you to determine Node Group members by Device Category, Vendor, Family, or Device Profile. If you configure multiple Device Filters, Nodes must match at least one of the Device Filters and pass any Additional Filters to belong to this Node Group.

Device Cat...	Device Vendor	Device Family
Switch-Ro...	Hewlett-Packard	HP ProCurve Switches

b Add specific nodes. Save and close the form.



Basics

Name	Important Nodes
Calculate Status	☐
Status	No Status
Add to View Filter List	☑

Notes

Important Nodes such as Edge Routers representing the egress points to local networks from WAN environments, and receive

Node hostnames listed here will belong to the Node Group regardless of any filters.

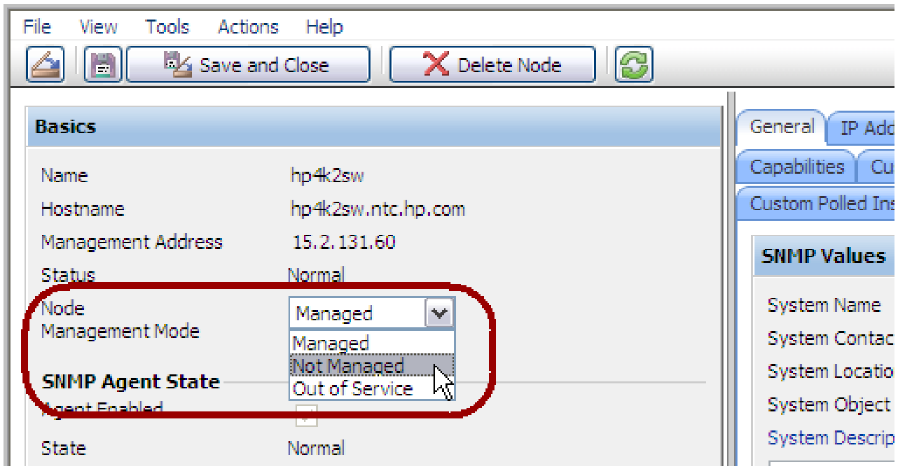
Node Hostname
router1.example.com
router7.example.com

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** form, as shown here:



Basics

Name	hp4k2sw
Hostname	hp4k2sw.ntc.hp.com
Management Address	15.2.131.60
Status	Normal
Node Management Mode	Managed

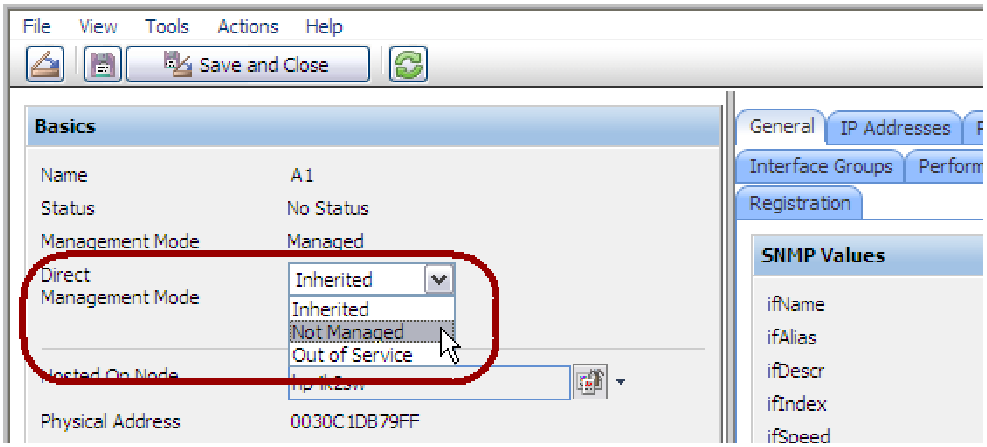
SNMP Agent State

Agent Enabled	
State	Normal

SNMP Values

System Name
System Contac
System Locatio
System Object
System Descrip

You can set the management mode of an individual interface on the **Interface** form, as shown here:



Basics

Name A1

Status No Status

Management Mode Managed

Direct Management Mode Inherited

Inherited

Not Managed

Out of Service

Hosted On Node hp1k23w

Physical Address 0030C1DB79FF

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 SNMPv2c format. Each event has an event object identifier, a name, and configuration parameters.

NNMi handles sources of events differently. Traps from devices and events forwarded from NNM management stations are in the SNMPv2c 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, see *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 with 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

It is recommended 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.ovpl` command, and then rerun `nnmicidentcfg.ovpl` with the `loadTraps` option.
   - To see which MIBs are already loaded, use the command:

     ```
     nnmloadmib.ovpl -list
     ```

For more information, see the *nnmincidentcfg.ovpl* and *nnmloadmib.ovpl* 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. The **SNMP Traps** tab displays the incidents configured for received SNMP traps.

4 Customize the trap incidents 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 Format**, 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 Events** 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 Format**, 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 by entering the following command:
   - *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 **Enabled** 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, see the *nnmtrapconfig.ovpl* reference page, or the UNIX manpage.

Configure Lifecycle Transition Actions

NNMi 9.0x Patch 2 (9.01) does not include up management event incidents. If you need notification that a node is up, associate a lifecycle transition action with the CLOSED lifecycle state of the NodeDown incident.

Integrations that use the NNMi northbound interface (including the NNMi Integration Module for Netcool Software), can receive traps that indicate when a NodeDown incident has been closed.

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 **Actions** tab).

   For most events, to match the behavior of NNM, set the **Lifecycle State** to **Registered**.

   For NNM up events, configure the corresponding NNMi down incident. For example, for the NNM NodeUp event, associate the action with the CLOSED lifecycle state of the NNMi NodeDown management event incident.

5 For each action script, verify the script functionality:

   - Does the script use parameters to input values from the incident? If so, update these parameters to the NNMi names. For the valid NNMi parameters, see *Valid Parameters for Configuring Incident Actions* in the NNMi help.
   - Does the script call any commands? If so, are these commands available on the NNMi management server, and do they produce the same output as on the NNM management station?

     For information about migrating NNM-provided commands to NNMi-provided commands, see Phase 7: Upgrade Custom Scripts on page 55.
   - Does the script logic work correctly on the NNMi management server?

Enhance in NNMi

6 Note the following NNMi configuration techniques:

   - You can configure more than one automatic action to occur when an event arrives (REGISTERED).
   - 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 launch actions that are available from the NNMi console menu.

Gather from NNM

1 Determine any custom operator actions in NNM.

Replicate to NNMi

2 For these custom actions, determine how to transfer them to be available as URLs.

   For a quick-reference list of all URL choices for launching NNMi, see **Help > Documentation Library > Integrate NNMi Elsewhere with URLs** in the NNMi console.

3 In the NNMi console, select **User Interface Configuration** from the **Configuration** workspace.

4 On the **Menu Items** tab, click **New**.

5. On the **Menu Item** form, enter the **Menu Item Label**, a **Unique Key**, **Ordering**, and **Selection Type**.
6. On the **Menu Item Contexts** tab, click **New**.
7. On the **Menu Item Context** form, for **Menu Item Action**, select **New Launch Action**.
8. On the **Launch Action** form, enter a **Name** and the **Full URL** for the action.
9. **Save and Close** back to the NNMi console.

Event Correlation: Repeating Events

NNM mechanisms use either the first or last event as the parent when deduplicating events.

NNMi creates a new parent with the **Dedup Stream Correlation** correlation nature. The parent incident appears in the **All Incidents** incident view. The original events appear in their configured incident views.

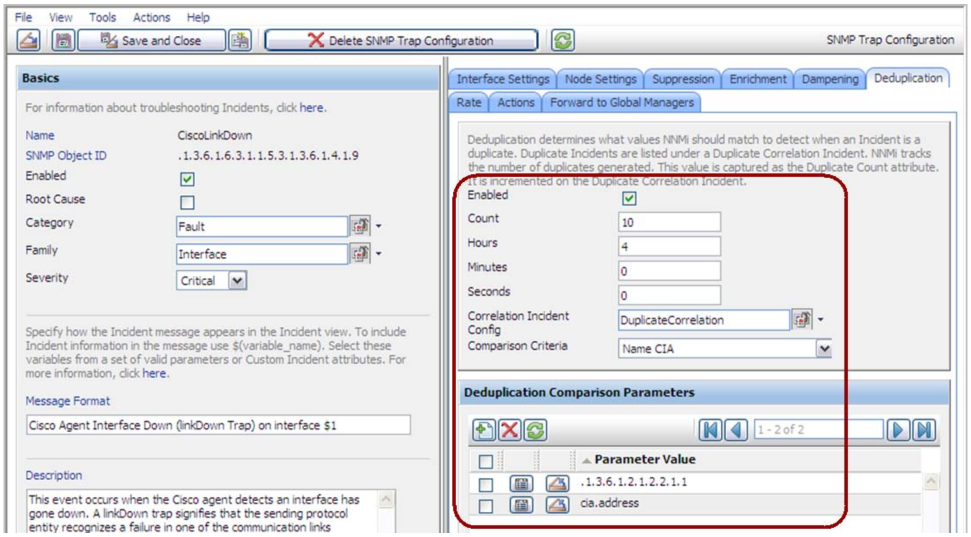
Gather from NNM

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 incident type to be deduplicated.
6. On the **Deduplication** tab, do the following:
   a. Select **Enabled** to enable monitoring.
   b. Set the count window.
   c. Set the time window (**Hours**, **Minutes**, and **Seconds** fields).
   d. Select **DuplicateCorrelation** as the new parent event (**Correlation Incident Config**).
   e. Define the **Comparison Criteria**.

For more information, see *Configure Deduplication for an SNMP Trap Incident* in the NNMi help.



Basics

For information about troubleshooting Incidents, click here.

Name: CiscoLinkDown
SNMP Object ID: .1.3.6.1.6.3.1.1.5.3.1.3.6.1.4.1.9
Enabled: ☑
Root Cause: □
Category: Fault
Family: Interface
Severity: Critical

Specify how the Incident message appears in the Incident view. To include Incident information in the message use $(variable_name). Select these variables from a set of valid parameters or Custom Incident attributes. For more information, click here.

Message Format: Cisco Agent Interface Down (linkDown Trap) on interface $1

Description: This event occurs when the Cisco agent detects an interface has gone down. A linkDown trap signifies that the sending protocol entity recognizes a failure in one of the communication links

Deduplication determines what values NNMi should match to detect when an Incident is a duplicate. Duplicate Incidents are listed under a Duplicate Correlation Incident. NNMi tracks the number of duplicates generated. This value is captured as the Duplicate Count attribute. It is incremented on the Duplicate Correlation Incident.

Enabled: ☑
Count: 10
Hours: 4
Minutes: 0
Seconds: 0
Correlation Incident Config: DuplicateCorrelation
Comparison Criteria: Name CIA

Deduplication Comparison Parameters

1 - 2 of 2

□	Parameter Value
□	.1.3.6.1.2.1.2.2.1.1
□	cia.address

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 with the **Rate Stream Correlation** correlation nature. The parent incident appears in the **All Incidents** incident view. The original events appear in their configured incident views. 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 Events** tab.
4. Open the incident type to be counted.
5. On the **Rate** tab, do the following:
   a. Select **Enabled** to enable monitoring.
   b. Set the count window.
   c. Set the time window (**Hours**, **Minutes**, and **Seconds** fields).
   d. Select **RateCorrelation** as the new parent event (**Correlation Incident Config**).
   e. Define the **Comparison Criteria**.

For more information, see *Configure Rate (Time Period and Count) for a Management Event Incident* 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. On the **Pairwise Configuration** tab, select an existing pair, or click **New**.
5. Configure the paired event identifiers and the matching criteria.

For more information, see *Pairwise Configuration Form* 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 Groups** 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 **Inventory** 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 > Management Mode > Out of Service**.
7. After maintenance is completed, select the nodes, and then select **Actions > Management Mode > 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 might 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 for can replicate into NNMi the location submap hierarchy of one OVW map. 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

1. Ensure that the upgrade tools have been set up as described in Phase 1: Collect Data from the NNM Management Station on page 21.
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** box.

If the upgrade 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 From the **Node Group** form for a replicated node group, click **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** form 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 might 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.

On NNMi management servers, the `nnmtopodump.ovpl` command located in the `bin` directory 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.

Gather from NNM

1 Copy all custom scripts for reading the NNM databases to a working directory.

Replicate in NNMi

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 replicating 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 with NNMi as two archive files (`migration.zip` for Windows operating systems, `migration.tar` for UNIX operating systems). After NNMi 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 2 lists the tools that are included in the data collection tools archive files.

Table 2 Upgrade Data Collection Tools

Tool	Description
createMigrationDirs.ovpl	Creates the directory structure to hold the upgrade data that will be collected from the NNM management station. For more information, see NNM Configuration Data Files on page 58.
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.

Table 2 Upgrade Data Collection Tools (cont'd)

Tool	Description
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 upgrade tools might use.
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 4.
trapdConfNodes.ovpl	Parses the `trapd.conf` file to create node lists (`EVENTS\NODES\*`) that future upgrade 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 3 lists the contents of the *<hostname>* directory.

Table 3 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
SNMP	Community strings
TOPO	A flat file of the topology database
WWW	The NNM web interface files

Data Import Tools for Upgrading

Table 4 lists the tools that NNMi provides for importing NNM 6.x/7.x data into the NNMi database. The upgrade process also uses standard NNMi tools. For information about the standard tools, see the appropriate reference pages, or the UNIX manpages.

Table 4 Data Import Tools

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 2.

Integrating NNM 6.x or NNM 7.x with NNMi

You can integrate the following HP Network Node Manager (NNM) 6.x/7.x functionality with HP Network Node Manager i Software (NNMi):

- You can forward events from NNM 6.x/7.x to the NNMi management server to use the NNMi incident views for managing incident life cycle.
- You can open some NNM 6.x/7.x views from the NNMi management server.

This integration is useful for controlling the rate of upgrading to NNMi.

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 throughout the network, you can maintain a few NNM 6.x/7.x management stations while using NNMi as your primary network management tool.

You can also use the information in this chapter to integrate a third-party product with NNMi. That product must be able to generate SNMP v1, v2c, or v3 traps and send them to the NNMi 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 management server, complete the following procedures in order:

- Step 1: Configure NNM 6.x/7.x to Forward Events to the NNMi 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 Topology
- Step 4: (Optional) Save the Management Station Configuration
- Step 5: Verify NNM 6.x/7.x Incident Configuration in the NNMi Console

Step 1: Configure NNM 6.x/7.x to Forward Events to the NNMi Management Server

On the NNM 6.x/7.x management station, configure each event that you want forwarded to the NNMi 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 Events** 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 64.

To configure an NNM 6.x/7.x event to forward to the NNMi 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 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 63.

- *Windows*: On the **Forwarding** tab in the **Modify Events** window, enter the host name of the NNMi management server in the **Destination** field.

  Click **Add**, and then click **OK**.

- *UNIX*: In the **Destination** field at the bottom of the **Event Configuration** window, enter the host name of the NNMi 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 all events you want to forward to an NNMi management server are configured.

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 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 Management Servers to receive events.
# This list should be small enough that it does not overwhelm the NNMi operators.
# In general, the events should be node-related, so that Neighbor Views launched remotely
from the NNMi 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 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 7.x, you can configure a node list for certain events. When a node list is present, an event coming into the NNM 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 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 management server.

For information about creating a node list in NNM 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 Topology

Include the NNM 6.x/7.x management station in the NNMi topology so that the NNMi 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 **Nodes** inventory view, add the management station to the discovery seeds, and then wait for it to be discovered.

For information about how to add a node to the discovery seeds, see *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 about these commands, see their respective reference pages, or the UNIX manpages.

Step 5: Verify NNM 6.x/7.x Incident Configuration in the NNMi Console

Verify that the events you forwarded from NNM 6.x/7.x are configured (as incidents) in NNMi.

To view the NNMi default incident configurations, follow these steps:

1. In the NNMi console, select **Incident Configuration** from the **Configuration** workspace.
2. Click the **Remote NNM 6.x/7.x Events** tab.

   This tab displays the default incident configurations.

The **Incident** form for this incident type shows the **Origin** of **NNM 6.x/7.x**.

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 Events** tab, add a new incident configuration for each missing event. For more information, see *Configuring Incidents* in the NNMi help.

➤ The incident categories in NNM 6.x/7.x are different from those in NNMi. For information about the relationship between the NNM 6.x/7.x alarm categories and the NNMi 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, the pre-configured incident categories are as follows:

- Accounting
- Application Status
- Configuration
- Fault
- Performance
- Security
- Status

Table 5 lists the mapping of NNM 6.x/7.x alarm categories to NNMi incident categories that HP suggests:

Table 5 Suggested Category Mappings

NNM 6.x/7.x Alarm Category	NNMi 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 management server to display NNM 6.x/7.x views on the NNMi management server, complete the following procedures in order:

- Step 1: Install Java Plug-in
- Step 2: Create an NNM 6.x/7.x Management Station Entity in NNMi
- Step 3: (Optional) Configure Additional NNM 6.x/7.x Views

Step 1: Install Java Plug-in

Although NNMi does not have any requirements for a Java Plug-in, NNM 6.x/7.x views require the use of a specific version of the Java Plug-in, which depends on the NNM version and operating system.

Review the latest release notes for your version of NNM, and then download and install the correct Java Plug-in version to all web browsers from which NNMi console users will launch NNM Dynamic Views.

Step 2: Create an NNM 6.x/7.x Management Station Entity in NNMi

Configure the NNMi management server to associate events received from the NNM 6.x/7.x management station to an entity in NNMi. This configuration enables the launching of NNM 6.x/7.x Dynamic Views from the NNMi management server. For example, you can select a Node Down from My7xSystem that is displayed in NNMi, 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 in NNMi, follow these steps:

1 In the NNMi console, select **Management Stations (6.x/7.x)** from the **Configuration** workspace.

2 Click **New.**

3 On the **Management Station** form, 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.
- **IP Address**—An IP address for the NNM 6.x/7.x management station. This IP address much be reachable from the NNMi 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 62 and if a configured event has been generated on the NNM 6.x/7.x management station and forwarded to the NNMi management server.
- **ovas 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.

4 Click **Save and Close**.

5 Sign out of the NNMi console.

The next time 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.example.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.example.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.example.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 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

Test 1: Verify Event Forwarding

Under normal network conditions, NNM 6.x/7.x generally receives network events. The NNM 6.x/7.x management stations forwards the configured events to NNMi, which displays them as remotely generated 6.x/7.x incidents. To expedite testing, you can generate a test event, or you can create an actual network failure on a test network or test device.

To verify event forwarding from the NNM 6.x/7.x management station to the NNMi 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 about how to run this command, see sendMsg.ovpl on page 71.

   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 70.

2. View the generated event in the NNMi console by selecting **NNM 6.x/7.x Events** from the **Incident Browsing** workspace.

   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 management server to see the list of events that the NNMi management server 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 might 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 in the NNM 6.x/7.x management station topology. 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 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
```

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 %COMPUTERNAME%"
```

- *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. To run this test, the OV_Message event in NNM 6.x/7.x must be configured to forward to the NNMi management server, and the OV_Message incident must be configured in the NNMi **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

1. In the NNMi 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** incident 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. See 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 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 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 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 Management Server on page 62.

6. Repeat step 5 until all expected events are included in the nnmdumpevents output on the NNMi management server.

7. In the NNMi 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 Events** tab of the **Incident Configuration** form.

Index

We appreciate your feedback!

If an email client is configured on this system, by default an email window opens when you click *here*.

If no email client is available, copy the information below to a new message in a web mail client, and then send this message to **ovdoc-nsm@hp.com**.

Product name and version: NNMi 9.0x Patch 2 (9.01)

Document title: *NNMi Upgrade Reference*

Feedback:

Java™
COMPATIBLE