# HP Network Node Manager i-series Software

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

Software Version: 8.10

## Deployment and Migration Guide

Document Release Date: November 2008
Software Release Date: November 2008

# Legal Notices

## Warranty

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

The information contained herein is subject to change without notice.

## Restricted Rights Legend

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

## Copyright Notices

© Copyright 2008 Hewlett-Packard Development Company, L.P.

This product includes software developed by the Apache Software Foundation (http://www.apache.org/). Portions Copyright © 1999-2003 The Apache Software Foundation. All rights reserved.

This product includes ASM Bytecode Manipulation Framework software developed by Institute National de Recherche en Informatique et Automatique (INRIA). Copyright © 2000-2005 INRIA, France Telecom. All Rights Reserved.

This product includes Commons Discovery software developed by the Apache Software Foundation (http://www.apache.org/). Copyright © 2002-2008 The Apache Software Foundation. All Rights Reserved.

This product includes Netscape JavaScript Browser Detection Library software, Copyright © Netscape Communications 1999-2001

This product includes Xerces-J xml parser software developed by the Apache Software Foundation (http://www.apache.org/). Copyright © 1999-2002 The Apache Software Foundation. All rights reserved.

This product includes software developed by the Indiana University Extreme! Lab (http://www.extreme.indiana.edu/). Xpp-3 Copyright © 2002 Extreme! Lab, Indiana University. All rights reserved.

## Trademark Notices

DOM4J® is a registered trademark of MetaStuff, Ltd.

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

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

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

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

UNIX® is a registered trademark of The Open Group.

## Oracle Technology — Notice of Restricted Rights

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

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

## Available Product Documentation

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

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

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

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

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

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

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

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

## Support

You can visit the HP software support web site at:

**www.hp.com/managementsoftware/services**

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

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

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

Most of the support areas require that you register as an HP Passport user and sign in. Many also require a support contract.

To find more information about access levels, go to:

**www.hp.com/managementsoftware/access_level**

To register for an HP Passport ID, go to:

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

# Contents


About This Guide . . . 15
What's in This Guide? . . . 15
Environment Variables Used in This Document . . . 16
Revision History . . . 17
For More Information about NNMi . . . 18

**Preparation** **19**

Hardware and Software Requirements . . . 21
Supported Hardware and Software . . . 21
System Configuration (UNIX) . . . 22
NNMi Coexistence with HP Performance Insight . . . 23

**Configuration** **25**

General Concepts for Configuration . . . 27
Task Flow Model . . . 27
Best Practice: Save the Existing Configuration . . . 28
Best Practice: Use the Author Attribute . . . 28
User Interface Model . . . 28
Ordering . . . 28
Node Groups . . . 29
Group Overlap . . . 30
Node Groups . . . 31
Hierarchies/Containment . . . 31
Device Filters . . . 32
Additional Filters . . . 32
Additional Nodes . . . 32
Interface Groups . . . 33
Node/Interface/Address Hierarchy . . . 33
Stop Everything and Start Over Again . . . 33

NNMi Communications . . . 35
Concepts for Communications . . . 35
Network Latency and Timeouts . . . 36
Access Credentials and Community Strings . . . 36
SNMP Version Preferences . . . 36
Configuration Levels . . . 37
Preferred Management Address . . . 38

Polling Protocols . . . 38
Plan Communications . . . 39
Meaningful Defaults . . . 39
Communication Configuration Regions . . . 39
Specific Node Configurations . . . 40
Retry and Timeout Values . . . 40
Active Protocols . . . 40
Multiple Community Strings or Access Credentials . . . 41
Configure Communications . . . 41
Evaluate Communications . . . 41
Are All Nodes Configured for SNMP? . . . 42
Is SNMP Access Currently Available for a Device? . . . 42
Is NNMi Using the Correct Communications Settings? . . . 42
Do the State Poller Settings Agree with the Communication Settings? . . . 43
Is the Management IP Address Correct? . . . 43
Tune Communications . . . 43

NNMi Discovery . . . 45
Concepts for Discovery . . . 45
NNMi Derives Attributes through Device Profiles . . . 46
Plan Discovery . . . 47
Select Your Primary Discovery Approach . . . 47
List-Based Discovery . . . 47
Rule-Based Discovery . . . 47
Auto-Discovery Rules . . . 48
Auto-Discovery Rule Ordering . . . 48
Exclude Devices from Discovery . . . 48
Ping Sweep . . . 49
Discovery Seeds for Auto-Discovery Rules . . . 49
Best Practices for Auto-Discovery Rules . . . 49
Examples . . . 49
Node Name Resolution . . . 50
Subnet Connection Rules . . . 51
Discovery Seeds . . . 51
Rediscovery Interval . . . 51
Do Not Discover Objects . . . 52
Configure Discovery . . . 52
Tips for Configuring Auto-Discovery Rules . . . 53
Tips for Configuring Seeds . . . 53
Evaluate Discovery . . . 54
Follow the Progress of Initial Discovery . . . 54
Were All Seeds Discovered? . . . 54
Do All Nodes Have a Valid Device Profile? . . . 55
Were All Nodes Discovered Properly? . . . 55
Auto-Discovery Rules . . . 55
IP Address Ranges . . . 55
System Object ID Ranges . . . 56
Are All Connections and VLANs Correct? . . . 56

Rediscover a Device . . . 56
Tune Discovery . . . 56

NNMi State Polling . . . 57
Concepts for State Polling . . . 57
Plan State Polling . . . 58
Polling Checklist . . . 58
What Can Be Monitored? . . . 59
Interfaces to Unmonitored Nodes . . . 60
Stop Monitoring . . . 61
Planning Groups . . . 61
Interface Groups . . . 62
Node Groups . . . 62
Planning Polling Intervals . . . 63
Deciding What Data to Collect . . . 64
Configure State Polling . . . 64
Configure Interface Groups and Node Groups . . . 64
Configure Interface Monitoring . . . 65
Configure Node Monitoring . . . 65
Verify Default Settings . . . 66
Evaluate State Polling . . . 66
Verify the Configuration for Network Monitoring . . . 66
Is the interface or node a member of the right group? . . . 66
Which settings are being applied? . . . 67
Which data is being collected? . . . 67
Evaluate the Performance of Status Polling . . . 67
Is the State Poller keeping up? . . . 68
Tune State Polling . . . 68

Administration 71

Enabling https for NNMi . . . 73
Obtain and Install a Public Key Certificate . . . 73
Self-Signed Certificate . . . 73
Third-Party Certificate . . . 74
Update the server.xml Files . . . 74
Communicate the New Connection Information to Users . . . 76

Configuring HP NNM i-series Software in a High Availability Cluster . . . 77
Supported HA Products . . . 78
Prerequisites to Configuring NNMi for HA . . . 78
HA Concepts . . . 79
HA Terms . . . 80
NNMi HA Cluster Scenarios . . . 80
Manpages . . . 83
Configuring HA . . . 84

Configuring NNMi for HA . . . 84
NNMi HA Configuration Information . . . 84
Configuring NNMi on the Primary Cluster Node . . . 86
Configuring NNMi on the Secondary Cluster Nodes . . . 88
Configuring iSPIs for HA . . . 89
iSPI HA Configuration Information . . . 89
Configuring the NNM iSPI for Performance as an Add-on iSPI . . . 89
Configuring an Add-On iSPI (All Other iSPIs) . . . 90
Configuring the NNM iSPI for Performance on a Dedicated Server for HA . . . 91
NNM iSPI for Performance HA Configuration Information . . . 91
Configuring the NNM iSPI for Performance on the Primary Cluster Node . . . 93
Configuring the NNM iSPI for Performance on the Secondary Cluster Nodes . . . 94
Configuring NNMi for HA in an Oracle Environment . . . 95
NNMi with Oracle HA Configuration Information . . . 95
Configuring NNMi with Oracle for HA . . . 95
Shared NNMi Data . . . 96
Data on the NNMi Shared Disk . . . 96
Configuration File Replication . . . 97
Configuring the Shared Disk . . . 97
A Note about Shared Disk Configuration on Windows 2003 . . . 98
Licensing NNMi in an HA Cluster . . . 98
Maintaining the HA Configuration . . . 99
Maintenance Mode . . . 99
Putting an HA Resource Group into Maintenance Mode . . . 99
Removing an HA Resource Group from Maintenance Mode . . . 99
Maintaining NNMi in an HA Cluster . . . 99
Starting and Stopping NNMi . . . 99
Changing NNMi Hostnames and IP Addresses in a Cluster Environment . . . 99
Stopping NNMi Without Causing Failover . . . 102
Restarting NNMi after Maintenance . . . 102
Maintaining Add-on iSPIs in an NNMi HA Cluster . . . 102
Maintaining the NNM iSPI for Performance on a Dedicated Server in an HA Cluster . . . 102
Starting and Stopping the NNM iSPI for Performance . . . 102
Changing the NNM iSPI for Performance System in a Cluster Environment . . . 103
Stopping NNM iSPI for Performance Without Causing Failover . . . 103
Restarting NNM iSPI for Performance after Maintenance . . . 103
Unconfiguring HA . . . 104
Unconfiguring NNMi from an HA Cluster . . . 104
Unconfiguring the NNM iSPI for Performance from an HA Cluster . . . 107
Upgrading NNMi under HA from NNMi 8.0x to NNMi 8.10 . . . 110
Troubleshooting the HA Configuration . . . 113
General HA Troubleshooting . . . 113
Error: Wrong Number of Arguments . . . 113
Product Startup Times Out (Solaris) . . . 113
Log Files on the Active Cluster Node Are Not Updating . . . 113
Cannot Start the HA Resource Group on a Particular Cluster Node . . . 114

NNMi-Specific HA Troubleshooting . . . 115
Re-Enable NNMi for HA after All Cluster Nodes are Unconfigured . . . 115
NNMi Does Not Start Correctly Under HA . . . 115
nmsdbmgr Does Not Start after HA Configuration . . . 116
pmd Does Not Start after HA Configuration . . . 116
Disk Failover Does Not Occur . . . 116
Shared Disk Files Are Not Found on the Secondary Node after Failover . . . 117
iSPI-Specific HA Troubleshooting . . . 117
iSPIs Do Not Start Correctly Under HA . . . 117
HA Configuration Reference . . . 118
NNMi HA Configuration Files . . . 118
NNM iSPI for Performance HA Configuration Files . . . 118
NNMi-Provided HA Configuration Scripts . . . 118
NNMi HA Configuration Log Files . . . 120
NNM iSPI for Performance HA Log Files . . . 121

NNMi Backup and Restore Tools . . . 123
About NNMi Data Backup . . . 123
Limitations of the NNMi Backup and Restore Scripts . . . 124
Backup and Restore Systems Must Be of the Same Configuration . . . 124
External Databases Are Not Supported . . . 124
Backing up NNMi Data . . . 125
Restoring NNMi Data . . . 127
Backing up and Restoring the Embedded Database Only . . . 128

Changing the NNMi Management Server . . . 131
Best Practices for Preparing the NNMi Configuration to Be Moved . . . 131
Moving the NNMi Configuration . . . 132
Changing the IP Address of an NNMi Management Server . . . 132

**Migration from NNM 6.x/7.x 133**

Product Comparison . . . 135
Network Discovery . . . 136
Key Concepts for Discovery . . . 138
Status Monitoring . . . 139
Key Concepts for Status Monitoring . . . 141
Customizing Event Monitoring . . . 142
Key Concepts for Event Monitoring . . . 144

Migrating from NNM 6.x/7.x . . . 145
Migration Options . . . 146
A Fresh Beginning . . . 146
Phased Migration . . . 146
Phase 1: Migrate SNMP Information . . . 148
Configure SNMP Access . . . 148
Limit Name Resolution . . . 151
Customize Device Profiles . . . 151

Phase 2: Migrate Discovery . . . 152
Schedule Discovery . . . 152
Select Your Discovery Method . . . 154
Configure Auto-Discovery Rules . . . 155
Configure Spiral Discovery . . . 155
Add Seeds to NNMi for Seeded Discovery . . . 159
Customize Connectivity . . . 160
Phase 3: Migrate Status Monitoring . . . 161
Set Polling Intervals . . . 161
Select Polling Protocol . . . 162
Configure Critical Nodes . . . 165
Exclude Objects from Status Polling . . . 166
Phase 4: Migrate Event Configuration and Event Reduction . . . 167
Display Traps from Devices . . . 167
Customize Display of NNMi-Generated Management Events . . . 169
Block/Ignore/Disable Traps . . . 169
Configure Automatic Actions . . . 170
Configure Additional (Manual) Actions . . . 170
Event Correlation: Repeating Events . . . 170
Event Correlation: Counting the Rate . . . 171
Event Correlation: Pairwise Cancellation . . . 172
Event Correlation: Scheduled Maintenance . . . 172

Integrating NNM 6.x or NNM 7.x with NNMi 8.10 . . . 173
Configure Event Forwarding . . . 174
Step 1: Configure NNM 6.x/7.x to Forward Events to the NNMi 8.10 Management Server . . . 174
Recommended and Supported Procedure: Use the Event Configuration Window . . . 174
Optional: Destination List File . . . 175
Alternative Procedure: Manually Edit trapd.conf . . . 176
Step 2: (Optional) Use Node Level Filtering to Further Reduce Events . . . 176
Step 3: Add the NNM 6.x/7.x Management Station to the NNMi 8.10 Topology . . . 176
Step 4: (Optional) Save the Management Station Configuration . . . 176
Step 5: Verify NNM 6.x/7.x Incident Configuration in the NNMi 8.10 Console . . . 177
Mapping Categories . . . 177
Configure Remote View Launching . . . 178
Step 1: Install Java Run-Time Environment (JRE) . . . 178
Step 2: Create an NNM 6.x/7.x Management Station Entity in NNMi 8.10 . . . 178
Step 3: (Optional) Configure Additional NNM 6.x/7.x Views . . . 179
URLs That Do Not Require a Selection . . . 180
URLs That Require a Selection . . . 180
Test the Integration . . . 180
Test 1: Verify Event Forwarding . . . 181
Generate Node/Interface Down/Up Events . . . 181
sendMsg.ovpl . . . 182
Test with Traps to NNM 6.x/7.x System . . . 182
Test 2: Launch NNM 6.x/7.x Dynamic Views from NNMi 8.10 . . . 183
Troubleshoot Event Forwarding . . . 183

## Integrations and Plug-ins 185


HP Operations Manager . . . 187
HP NNMi–HPOM Integration . . . 187
Value . . . 188
Supported Versions . . . 189
Documentation . . . 189
Enabling the HP NNMi–HPOM Integration . . . 189
HPOM for Windows . . . 190
HPOM for UNIX . . . 191
Using the HP NNMi–HPOM Integration . . . 193
Changing the NNMi–HPOM Integration Configuration . . . 194
Disabling the HP NNMi–HPOM Integration . . . 194
For All HPOM Management Servers . . . 194
For One HPOM Management Server . . . 195
Troubleshooting the HP NNMi–HPOM Integration . . . 195
HPOM Does Not Receive Any Forwarded Incidents . . . 195
HPOM Does Not Receive Some Forwarded Incidents . . . 197
NNMi Incident Information Is Not Available in the HPOM Messages Browser . . . 197
NNMi and HPOM Are Not Synchronized . . . 197
The Integration Does Not Work Through a Firewall . . . 197
HP NNMi–HPOM Integration Configuration Form Reference . . . 198
NNMi Management Server Connection . . . 198
HPOM Management Server Connection . . . 199
Integration Behavior . . . 200
Incident Filters . . . 200
Example Incident Filters . . . 202
Incident Filter Limitations . . . 203
HP NNMi–HPOM Integration Log File Reference . . . 204
Log Files . . . 204
Enabling More Detailed Messages in the nnm-om Log File . . . 204
Enabling Integration Communication Messages in the jboss-log4j Log File . . . 205
Example Log File Messages . . . 205
Good Start up Trace . . . 206
NNMi Incident Successfully Forwarded to HPOM for UNIX . . . 207
NNMi Incident Successfully Closed in HPOM for UNIX . . . 207
Incorrect HPOM Login Configuration . . . 207
Incorrect NNMi Login Configuration . . . 208
Acknowledgement Error from HPOM for Windows . . . 209
Acknowledgement Error from HPOM for UNIX . . . 210
Node Not Found Error . . . 212

HP Universal Configuration Management Database . . . 213
HP NNMi–HP UCMDB Integration . . . 213
Value . . . 214
Supported Versions . . . 214
Documentation . . . 214
Enabling the HP NNMi–HP UCMDB Integration . . . 214

Using the HP NNMi–HP UCMDB Integration . . . 215
Viewing Impacted CIs. . . . 215
Viewing the UCMDB CI . . . 216
Changing the HP NNMi–HP UCMDB Integration Configuration . . . 216
Disabling the HP NNMi–HP UCMDB Integration . . . 217
Troubleshooting the HP NNMi–HP UCMDB Integration. . . . 217
HP NNMi–HP UCMDB Integration Configuration Form Reference . . . 217
NNMi Management Server Connection . . . 218
UCMDB Server Connection . . . 219
Integration Behavior . . . 219

AlarmPoint . . . 221
HP NNMi–AlarmPoint Integration. . . . 221
Value . . . 222
Supported Versions . . . 222
Documentation . . . 222
Enabling the HP NNMi–AlarmPoint Integration . . . 223
Using the HP NNMi–AlarmPoint Integration . . . 223
Disabling the HP NNMi–AlarmPoint Integration . . . 224
Troubleshooting the HP NNMi–AlarmPoint Integration . . . 224

nGenius Performance Manager . . . 225
HP NNMi–nGenius Performance Manager Integration . . . 226
Value . . . 226
Supported Versions . . . 226
Documentation . . . 227
Enabling the HP NNMi–nGenius Performance Manager Integration . . . 227
Using the HP NNMi–nGenius Performance Manager Integration. . . . 227
Disabling the HP NNMi–nGenius Performance Manager Integration. . . . 228
Troubleshooting the HP NNMi–nGenius Performance Manager Integration . . . 228

NNM iSPI for Performance . . . 229
What is the NNM iSPI for Performance? . . . 229
Value . . . 230
Supported Versions . . . 230
Documentation . . . 230
Enabling the NNM iSPI for Performance. . . . 230
Configuring the NNM iSPI for Performance to Run Under HA . . . 231
Using the NNM iSPI for Performance . . . 231
Backup Provisions for the NNM iSPI for Performance . . . 231
Resetting the NNM iSPI for Performance . . . 232
Moving the NNM iSPI for Performance to a Different Computer. . . . 232
Disabling the NNM iSPI for Performance . . . 232
Troubleshooting the NNM iSPI for Performance . . . 232

## Additional Information 233


NNMi Environment Variables . . . . . 235

Environment Variables Used in This Document . . . . . 235

Other Available Environment Variables. . . . . 236

Windows Paths and Environment Variables from NNMi 8.00 . . . . . 238

The Causal Engine and NNMi Incidents . . . . . 241

Causality Analysis - High Level Discussion. . . . . 241

Causal Engine Concepts . . . . . 242

Concept of Status . . . . . 242

What is an Episode?. . . . . 243

What Does NNMi Analyze?. . . . . 243

What Are the Failure Scenarios? . . . . . 246

SNMP Agent Not Responding to SNMP Queries . . . . . 247

SNMP Agent Responding to SNMP Queries. . . . . 248

IPv4Address Not Responding to ICMP . . . . . 249

IPv4Address Responding to ICMP. . . . . 250

Interface is Operationally Down . . . . . 251

Interface is Operationally Up. . . . . 252

Interface is Administratively Down. . . . . 253

Interface is Administratively Up . . . . . 254

Connection is Operationally Down . . . . . 255

Connection is Operationally Up. . . . . 256

Directly Connected Node is Down . . . . . 257

Directly Connected Node is Up . . . . . 258

Indirectly Connected Node is Down. . . . . 259

Indirectly Connected Node is Up . . . . . 260

Directly Connected Node is Down and Creates a Shadow . . . . . 261

Directly Connected Node is Up, Clearing the Shadow . . . . . 262

Important Node is Unreachable. . . . . 263

Important Node is Reachable. . . . . 263

Node or Connection is Down . . . . . 264

Node or Connection is Up. . . . . 264

Island Group is Down. . . . . 265

Island Group is Up . . . . . 266

Network Configuration Changes. . . . . 267

Node Updated . . . . . 267

Interface Moves to and from Connections . . . . . 267

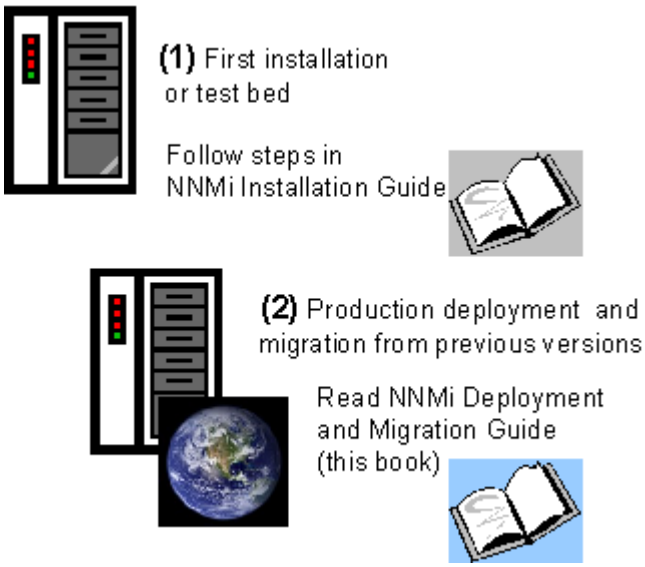
Device Generated Traps . . . . . 267

NNMi Management Configuration Changes . . . . . 268

Glossary . . . . . 269

Index . . . . . 277

# About This Guide




This chapter contains the following topics:

- What's in This Guide?
- Environment Variables Used in This Document
- Revision History
- For More Information about NNMi

## What's in This Guide?

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

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

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

# Environment Variables Used in This Document

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

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

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

➤ If you first installed NNMi 8.00, your system uses different values for these environment variables, as described in Windows Paths and Environment Variables from NNMi 8.00 on page 238.

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

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

For information on other NNMi environment variables that you can source, see NNMi Environment Variables on page 235.

# Revision History

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

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

# For More Information about NNMi

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

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

| What do you want to do? | Where to find more information |
|---|---|
| Install NNMi or NNMi Advanced (first time). | Download the *NNMi Installation Guide*. This guide contains basic steps to install and un-install the product, plus how to do an initial configuration using the NNMi Quick Start Configuration Wizard.<br>• *HP Network Node Manager i-series Software Installation Guide for the Windows Operating System*<br>• *HP Network Node Manager i-series Software Installation Guide for the HP-UX Operating System*<br>• *HP Network Node Manager i-series Software Installation Guide for the Linux Operating System*<br>• *HP Network Node Manager i-series Software Installation Guide for the Solaris Operating System* |
| Plan for network deployment, including links to system requirements. | See Preparation on page 19 of this guide. |
| Configure NNMi for a production environment. | See Configuration on page 25 of this guide. |
| Administer NNMi, including configuration for high availability (HA) or backup. | See Administration on page 71 of this guide. |
| Migrate to NNMi from previous versions of Network Node Manager. | See Migration from NNM 6.x/7.x on page 133 of this guide. |
| Learn more about products that integrate with NNMi, including NNMi iSPIs. | See Integrations and Plug-ins on page 185 of this guide. |
| Install the HP NNMi iSPI Network Engineering Toolset (iNET) and learn about its components. | Download the *HP Network Node Manager i-series Software iSPI Network Engineering Toolset Planning and Installation Guide*. |
| Obtain documentation for the NNM Developer's Toolkit (SDK). | Download the *NNMi Developer's Toolkit Guide*. |

# Preparation

This section contains the following chapter:

- Hardware and Software Requirements

# Hardware and Software Requirements

This chapter contains the following topics:

- Supported Hardware and Software
- System Configuration (UNIX)
- NNMi Coexistence with HP Performance Insight

## Supported Hardware and Software

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

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

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

**Table 1 Software and hardware pre-installation checklist**

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

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

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

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

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

# System Configuration (UNIX)

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

# NNMi Coexistence with HP Performance Insight

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

1. Install HP Performance Insight first.

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

2. Stop all HP Performance Insight processes.
3. Install NNMi. Refer to the *NNMi Installation Guide* for specific instructions.
4. Stop all NNMi processes:

   **`ovstop -c`**

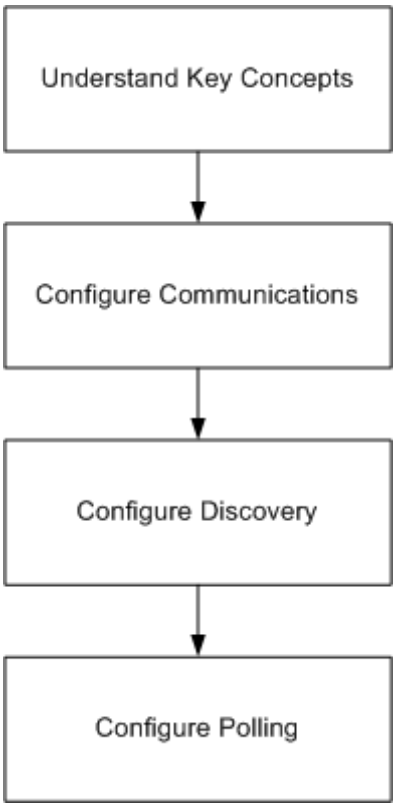
5. Modify the `nnm.port.properties` file to resolve any port conflicts. You can find this file in the following directory:
   - *Windows*: `%NnmDataDir%\shared\nnm\conf\`
   - *UNIX*: `$NnmDataDir/shared/nnm/conf`
6. Start HP Performance Insight processes.
7. Start all NNMi processes:

   **`ovstart -c`**

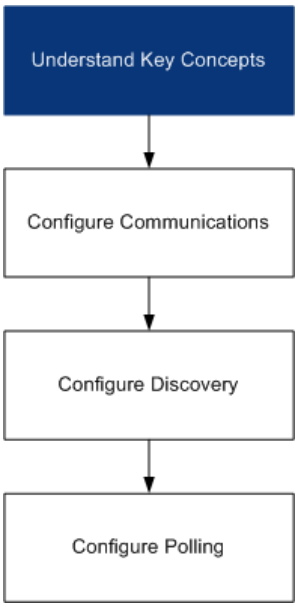
# Configuration

This section contains the following chapters:

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



# General Concepts for Configuration




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

This chapter contains the following topics:


- Task Flow Model
- Best Practice: Save the Existing Configuration
- Best Practice: Use the Author Attribute
- User Interface Model
- Ordering
- Node Groups
- Node/Interface/Address Hierarchy
- Stop Everything and Start Over Again


## Task Flow Model

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

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

## Best Practice: Save the Existing Configuration

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

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

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

## Best Practice: Use the Author Attribute

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

- Device Profile
- Incident Configuration
- URL Action

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

## User Interface Model

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

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

## Ordering

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

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

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

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

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

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

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

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

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

# Node Groups

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

- Monitoring settings
- Table filtering
- Customizing map views

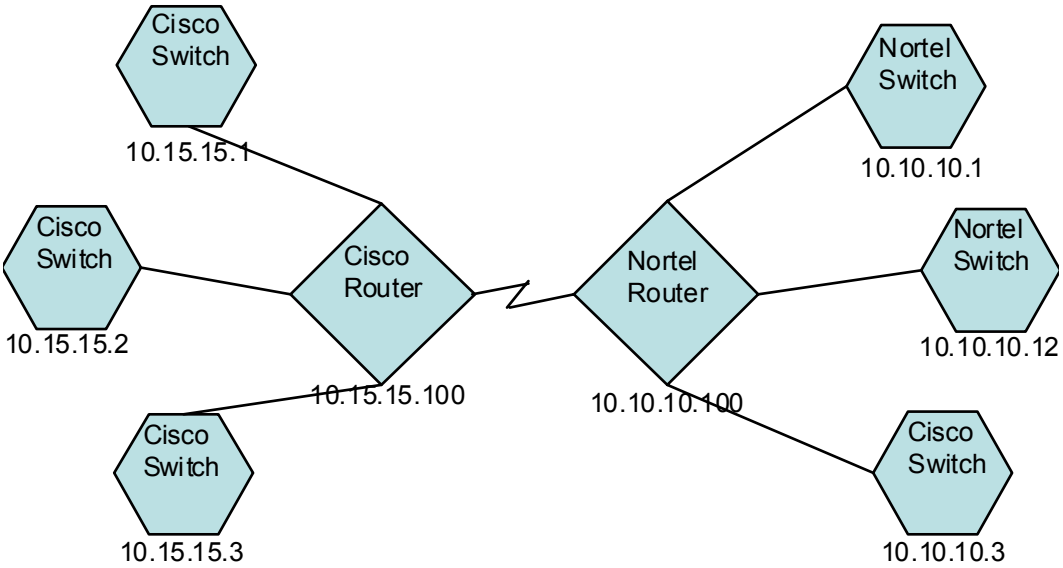
Interface groups can be used for either or both of the following purposes:

- Monitoring settings
- Table filtering

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

## Group Overlap

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




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

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

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

## Node Groups

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

- Additional nodes that you specify explicitly

OR'd with

- Inclusion of child groups

OR'd with

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

  AND with

  — Additional filters based on other attributes

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

### Hierarchies/Containment

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

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

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

Hierarchies work well for the following situations:

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

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

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

The configuration interface automatically prevents circular hierarchy definitions.

## Device Filters

During discovery, NNMi collects direct information through SNMP queries and derives other information from that through device profiles. (For more information, see NNMi Derives Attributes through Device Profiles on page 46.) By gathering the system object ID, NNMi can index through the correct device profile to derive the following information:

- Vendor
- Device category
- Device family within the category

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

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

## Additional Filters

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

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

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

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

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

## Additional Nodes

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

### Interface Groups

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

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

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

## Node/Interface/Address Hierarchy

NNMi assigns monitoring settings in the following manner:

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

Child node groups are included in the ordering hierarchy. If the parent node group has a lower ordering number (for example, parent=10, child=20), then the monitoring configuration specified for the parent node group also applies to the nodes in the child node group. To override a parent node group monitoring configuration, set the ordering number for the child node group to a number that is lower than the parent (for example, parent=20, child=10).

3. **Default Settings**—If no match is found for a node, interface, or IP address in step 1 or step 2, NNMi applies the default monitoring configuration settings.

## Stop Everything and Start Over Again

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

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

Follow these steps:

1. Stop the NNMi services:

```
ovstop -c
```

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

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

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

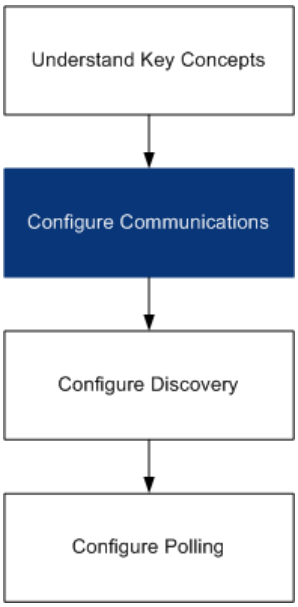
     **`nnmresetembdb.ovpl -nostart`**

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

   **`ovstart -c`**

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

# NNMi Communications




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

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

This chapter contains the following topics:


- Concepts for Communications
- Plan Communications
- Configure Communications
- Evaluate Communications
- Tune Communications


## Concepts for Communications

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

The following concepts apply to communications configuration:


- Network Latency and Timeouts
- Access Credentials and Community Strings
- SNMP Version Preferences
- Configuration Levels

- Preferred Management Address
- Polling Protocols

## Network Latency and Timeouts

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

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

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

When NNMi goes through its set of retries, it doubles the previous timeout value for each iteration.

## Access Credentials and Community Strings

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

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

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

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

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

## SNMP Version Preferences

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

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

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

## Configuration Levels

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

- Specific node
- Regions
- Global Defaults

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

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

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

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

## Preferred Management Address

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

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

Best practice

One of NNMi's rules specifies using the first discovered IP address as the management address. You can influence NNMi's choosing by configuring your preferred IP as the seed address. For suggestions regarding seed selection, see Discovery Seeds on page 51.

## Polling Protocols

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

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

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

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

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

# Plan Communications

Make decisions in the following areas:

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

## Meaningful Defaults

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

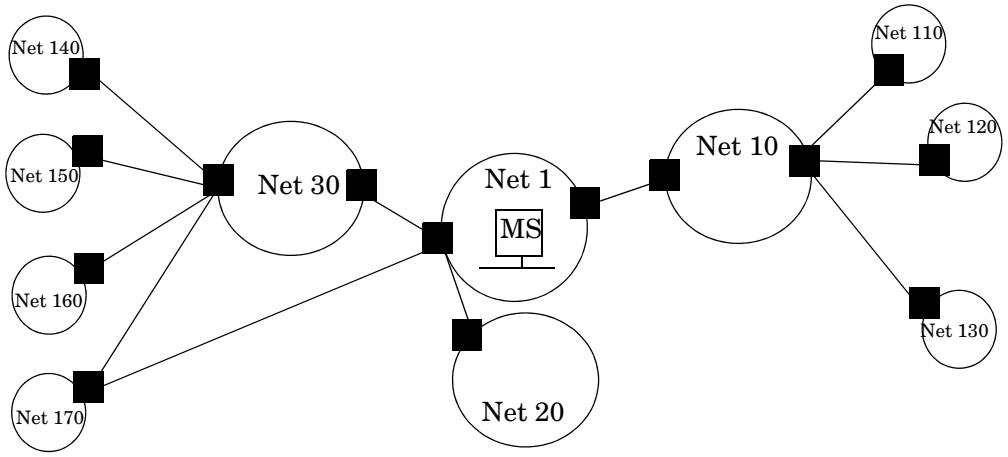
- Do you have commonly-used community strings that NNMi should try?
- If no other settings apply, what timeout and retry values are reasonable in your network?

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

## Communication Configuration Regions

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

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



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

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

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

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

## Specific Node Configurations

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

## Retry and Timeout Values

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

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

## Active Protocols

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

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

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

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

### Multiple Community Strings or Access Credentials

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

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

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

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

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

## Configure Communications

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

It is a good idea to save a copy of the existing configuration before you make any major configuration changes. For more information, see Best Practice: Save the Existing Configuration on page 28.

Configure the following areas of communication:

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

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

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

Best practice

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

## Evaluate Communications

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

Consider the following:

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

## Are All Nodes Configured for SNMP?

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

## Is SNMP Access Currently Available for a Device?

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

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

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

## Is NNMi Using the Correct Communications Settings?

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

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

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

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

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

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

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

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

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

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

   The form shows the protocol version and other SNMP settings.

### Do the State Poller Settings Agree with the Communication Settings?

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

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

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

### Is the Management IP Address Correct?

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

## Tune Communications

**Reduce authentication failures**

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

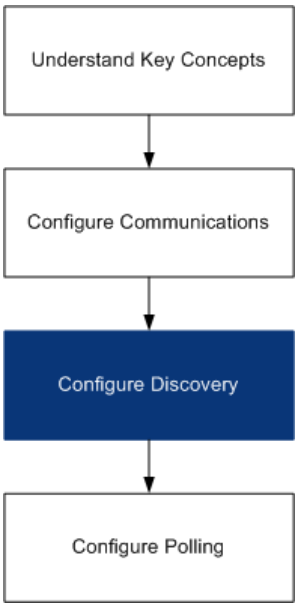
**Tune timeouts and retries**

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

**Reduce default community strings**

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

# NNMi Discovery




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

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

If you have experience working with NNM 6.x/7.x and you want to understand how discovery has changed in NNMi 8.10, see Network Discovery on page 136 for a high-level overview of the differences.

This chapter contains the following topics:


- Concepts for Discovery
- Plan Discovery
- Configure Discovery
- Evaluate Discovery
- Tune Discovery


## Concepts for Discovery

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

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

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

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

Status monitoring considerations might also influence your choices. By default, the State Poller only monitors interfaces connected to devices NNMi has discovered. You can override this default for some areas of your network, and you can discover the devices beyond the edge of your responsibility. (For information on the State Poller, see NNMi State Polling on page 57.)

NNMi provides two primary discovery configuration models:

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

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

## NNMi Derives Attributes through Device Profiles

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

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

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

# Plan Discovery

Make decisions in the following areas:

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

## Select Your Primary Discovery Approach

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

### List-Based Discovery

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

Benefits of using only list-based discovery include:

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

Disadvantages of using only list-based discovery include:

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

### Rule-Based Discovery

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

Benefits of using rule-based discovery include:

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

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

Disadvantages of using rule-based discovery include:

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

Rule-based discovery only

## Auto-Discovery Rules

### Auto-Discovery Rule Ordering

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

- IP address ranges

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

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

### Exclude Devices from Discovery

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

## Ping Sweep

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

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

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

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

Best practice

Enable ping sweep for small discovery ranges only.

## Discovery Seeds for Auto-Discovery Rules

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

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

## Best Practices for Auto-Discovery Rules

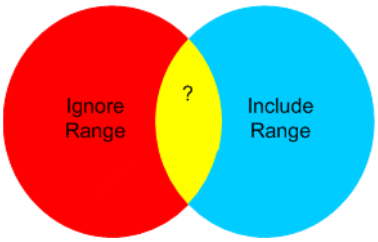
- Because NNMi automatically manages all discovered devices, use IP address ranges that closely match the areas of the network that you want to manage.
  - — You can use multiple IP address ranges within an auto-discovery rule to restrict discovery.
  - — You can add a large IP address range to an auto-discovery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

## Examples

### Discovery Rule Overlap

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

**Figure 1 Overlapping Discovery Ranges**




### Limit Device Type Discovery

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

## Node Name Resolution

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

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

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

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

Best practice

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

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

Best practice

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

## Subnet Connection Rules

List-based discovery only

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

Rule-based discovery only

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

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

Best practice

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

## Discovery Seeds

List the devices to use as discovery seeds.

Best practice

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

Best practice

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

List-based discovery only

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

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

Rule-based discovery only

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

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

Best practice

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

## Rediscovery Interval

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

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

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

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

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

### Do Not Discover Objects

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

- On the **Communication Configuration** form, you can turn off ICMP and/or SNMP communication at various levels: globally, for communication regions, or for specific hostnames or IP addresses. For information on the impacts of disabling one or both of these protocols, see Polling Protocols on page 38.
- On the **Discovery Configuration** form, you can set up an auto-discovery rule that instructs NNMi to never gather hints from certain IP addresses or SNMP system object IDs. Nodes matching the criteria still appear on the map and in the database, but spiral discovery does not extend to the neighboring devices beyond those IP addresses or object types.
- On the **Discovery Configuration** form, you can set up an auto-discovery rule that instructs NNMi to exclude certain IP address ranges and/or specific IP addresses from the database. Spiral discovery does not display those addresses on any node's list of addresses or use those addresses when establishing connections between devices, so NNMi never monitors the health of those addresses.

## Configure Discovery

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

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

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

It is a good idea to save a copy of the existing configuration before you make any major configuration changes. For more information, see Best Practice: Save the Existing Configuration on page 28.

## Tips for Configuring Auto-Discovery Rules

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

## Tips for Configuring Seeds

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

  ```
  IP_address # node name
  ```

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

Rule-based discovery only

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

# Evaluate Discovery

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

## Follow the Progress of Initial Discovery

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

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

## Were All Seeds Discovered?

1. Open the **Discovery Configuration** form.
2. On the **Discovery Seeds** tab, sort the list of nodes by the **Discovery Seed Results** column. For any node in an error state, consider the following:
   - Failed discovery due to an unreachable node or unresolved DNS name—For these types of failures, verify network connectivity to the node and check for accurate DNS name resolution. To work around DNS issues, use the IP address to seed the node or include the hostname in a `hostNoLookup.conf` file.
   - License node count exceeded—This scenario occurs when the number of devices already discovered reached your license limit. You can either delete some discovered nodes or purchase additional node pack licenses.
   - Node discovered but no SNMP response—SNMP communication problems can occur for seeded devices as well as devices that are discovered through auto-discovery. For more information, see Evaluate Communications on page 41.

## Do All Nodes Have a Valid Device Profile?

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

## Were All Nodes Discovered Properly?

Examine the data in the **Nodes** inventory view. If any nodes do not have a management address, check the communication settings for those nodes as described in Are All Nodes Configured for SNMP? on page 42.

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

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

List-based discovery only

## Auto-Discovery Rules

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

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

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

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

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

### IP Address Ranges

The IP address of a discovery hint must match an **Include in Rule** entry in the IP address range list. If there are no included IP address ranges in an auto-discovery rule, then all address hints are considered a match. (For this case, see Tips for Configuring Auto-Discovery Rules on page 53.) Additionally, the hint must *not* match any entry marked **Ignored by Rule**. If all checks successfully match, this rule's configuration is used for handling the hint.

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

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

### System Object ID Ranges

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

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

## Are All Connections and VLANs Correct?

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

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

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

## Rediscover a Device

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

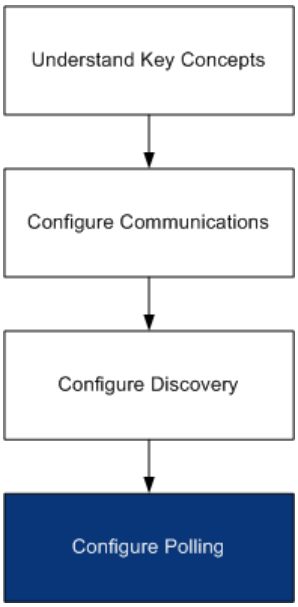
If the device is a seed, delete the seed, and then re-add the seed.

# Tune Discovery

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

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

# NNMi State Polling




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

If you have experience working with NNM 6.x/7.x and you want to understand how monitoring has changed in NNMi 8.10, see Status Monitoring on page 139 for a high-level overview of the differences.

This chapter contains the following topics:

- Concepts for State Polling
- Plan State Polling
- Configure State Polling
- Evaluate State Polling
- Tune State Polling

## Concepts for State Polling

This section provides a brief overview of network monitoring, including the order that the State Poller uses to evaluate polling groups. After reading the information in this section, continue to Plan State Polling on page 58 for more specific information.

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

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

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

As you plan polling configuration, you should carefully consider how to set up interface groups and node groups for the State Poller service. If you are new to the concept of *groups*, see Node Groups on page 29, and Node/Interface/Address Hierarchy on page 33 for overview information.

**Order of evaluation**

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

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

# Plan State Polling

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

## Polling Checklist

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

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

Example polling configuration

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

1. What can be monitored?

   Because you can only monitor what has been discovered, you configure auto-discovery rules to ensure that NNMi's database contains your ProximiT proxy servers. For more information on configuring discovery, see NNMi Discovery on page 45.

2. What are the logical groups for monitored items?

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

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

3. How often should each group be monitored?

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

4. What data should be collected?

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

For more detailed planning information concerning these configuration choices, see the sections below:


— What Can Be Monitored? on page 59

— Planning Groups on page 61

— Planning Polling Intervals on page 63

— Deciding What Data to Collect on page 64


## What Can Be Monitored?

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

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

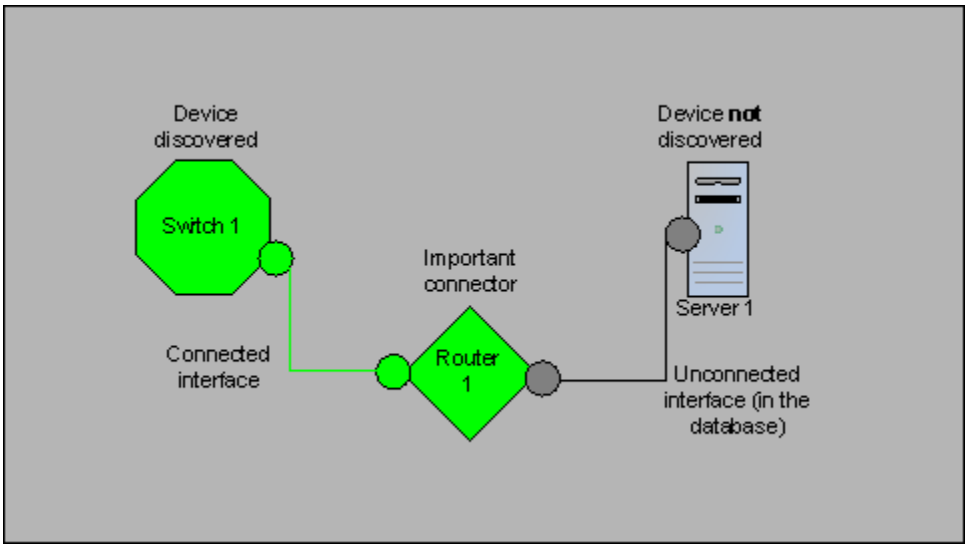
In most cases, polling only connected interfaces provides sufficiently accurate root-cause analysis. Extending the set of monitored interfaces can impact polling performance.

## Extend monitoring

You can extend the monitoring to include the following:

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

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




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

### Interfaces to Unmonitored Nodes

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

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

- Discover the unmonitored node

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

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

- Poll the unconnected interface

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

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

### Stop Monitoring

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

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

Best practice

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

## Planning Groups

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

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

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

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

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

Best practice

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

## Interface Groups

Based on your criteria, determine which Interface groups to create. Remember that interface groups are evaluated first (see Concepts for State Polling on page 57). Interface groups can reference node group membership, so you may end up configuring node groups before interface groups to implement your plan.

**Preconfigured interface groups**

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

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

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

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

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

## Node Groups

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

### Preconfigured node groups

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

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

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

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

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

Best practice

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

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

### Interaction with Device Profiles

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

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

## Planning Polling Intervals

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

Best practice

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

## Deciding What Data to Collect

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

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

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

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

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

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

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

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

# Configure State Polling

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

It is a good idea to save a copy of the existing configuration before you make any major configuration changes. For more information, see Best Practice: Save the Existing Configuration on page 28.

## Configure Interface Groups and Node Groups

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

Examples

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

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

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

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

## Configure Interface Monitoring

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

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

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

Best practice

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

## Configure Node Monitoring

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

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

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

You may configure different settings for each node group.

Best practice

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

### Verify Default Settings

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

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

## Evaluate State Polling

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

### Verify the Configuration for Network Monitoring

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

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

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

- Node Groups
- Interface Groups

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

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

If the object is not a member of a group:

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

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

### Which settings are being applied?

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

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

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

### Which data is being collected?

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

## Evaluate the Performance of Status Polling

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

### Is the State Poller keeping up?

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

**Table 2 State Poller Health Information**

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

## Tune State Polling

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

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

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

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

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

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

# Administration

This section contains the following chapters:


- Enabling https for NNMi
- Configuring HP NNM i-series Software in a High Availability Cluster
- NNMi Backup and Restore Tools
- Changing the NNMi Management Server

# Enabling https for NNMi

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

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

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

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

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

## Obtain and Install a Public Key Certificate

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

### Self-Signed Certificate

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

To obtain and install a self-signed certificate, follow these steps:

1 Use the Java `keytool` generator to create the certificate file.

- To use the command line tool, run a command similar to the following example:

```
keytool -genkey -keystore "c:\path\to\my\.keystore"
-storepass changeit -keypass changeit -keyalg RSA
-validity 3650 -alias alias
```

- To use the interactive tool, run the following command:

```
keytool -genkey
```

2 Move the certificate file to the location that you specified while running the `keytool` command.

### Third-Party Certificate

A third-party certificate is created and maintained by a business that can ascertain that the certificate is being used only by the purchaser of that certificate. This business relationship builds trust that the web server is offered by a valid entity. If your organization already has a third-party certificate, you can use that certificate for the NNMi web server.

To obtain and install a third-party certificate, follow the instructions in the Tomcat configuration help that is referenced earlier in this chapter.

## Update the server.xml Files

The version of jboss that is delivered with NNMi 8.10 includes the Tomcat server version 6.0. The https configuration process for Tomcat 6.0 is different from the process for previous versions of Tomcat.

The Tomcat server is configured in the following file:

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

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

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

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

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

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

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

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

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

If you want users inside the Intranet to connect to the NNMi console with the http protocol, leave this default block in place. If you want all users to use the https protocol for connecting to the NNMi console, delete or comment out this block.

To specify the https protocol, uncomment the appropriate https connector block and modify it as appropriate.

After modifying the `server.xml` file, restart the NNMi services:

```
ovstop
ovstart
```

- If you are using a self-signed certificate, use the first connector block in the `<!-- HTTPS/SSL CONFIG -->` section of the `server.xml` file, which is similar to the following example. In this block, the bold lines are those that identify the self-signed certificate.

```
<Connector port="${jboss.https.port}"
    address="${jboss.bind.address}"
    maxThreads="250"
    acceptCount="100"
    connectionTimeout="20000"
    strategy="ms"
    maxHttpHeaderSize="8192"
    emptySessionPath="true"
    scheme="https"
    secure="true"
    SSLEnabled="true"
    keystoreFile="${path-to-file}\.keystore-filename"
    keystorePass="changeit"
    keyAlias="alias"
    clientAuth="false"
    sslProtocol="TLS" />
```

- If you are using a third-party certificate, use the second connector block in the `<!-- HTTPS/SSL CONFIG -->` section of the `server.xml` file, which is similar to the following example. In this block, the bold lines are those that identify the third-party certificate.

```
<Connector port="${jboss.https.port}"
    address="${jboss.bind.address}"
    maxThreads="250"
    acceptCount="100"
    connectionTimeout="20000"
    strategy="ms"
    maxHttpHeaderSize="8192"
    emptySessionPath="true"
    scheme="https"
    secure="true"
    SSLEnabled="true"
    SSLCertificateFile="/usr/local/ssl/server.crt"
    SSLCertificateKeyFile="/usr/local/ssl/server.pem"
    clientAuth="false"
    sslProtocol="TLS" />
```

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

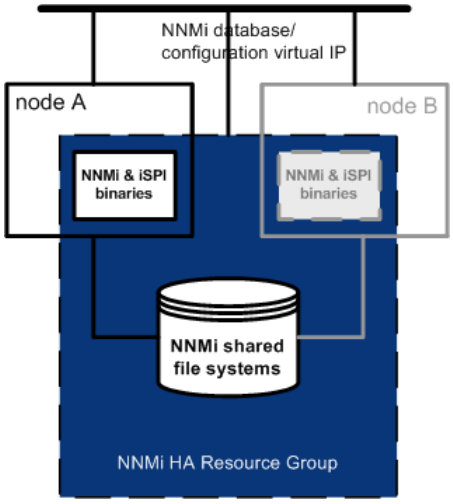
## Communicate the New Connection Information to Users

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

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

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

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




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

As of HP Network Node Manager i Software version 8.02, NNMi provides support for configuring NNMi to run in an HA cluster under one of several separately-purchased HA products. As of NNMi version 8.10, the NNMi Smart Plug-ins (iSPIs) can also be running under HA.

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

This chapter contains the following topics:


- Supported HA Products
- Prerequisites to Configuring NNMi for HA
- HA Concepts
- Configuring HA
- Shared NNMi Data
- Licensing NNMi in an HA Cluster
- Maintaining the HA Configuration
- Unconfiguring HA
- Upgrading NNMi under HA from NNMi 8.0x to NNMi 8.10
- Troubleshooting the HA Configuration
- HA Configuration Reference

# Supported HA Products

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

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

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

# Prerequisites to Configuring NNMi for HA

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

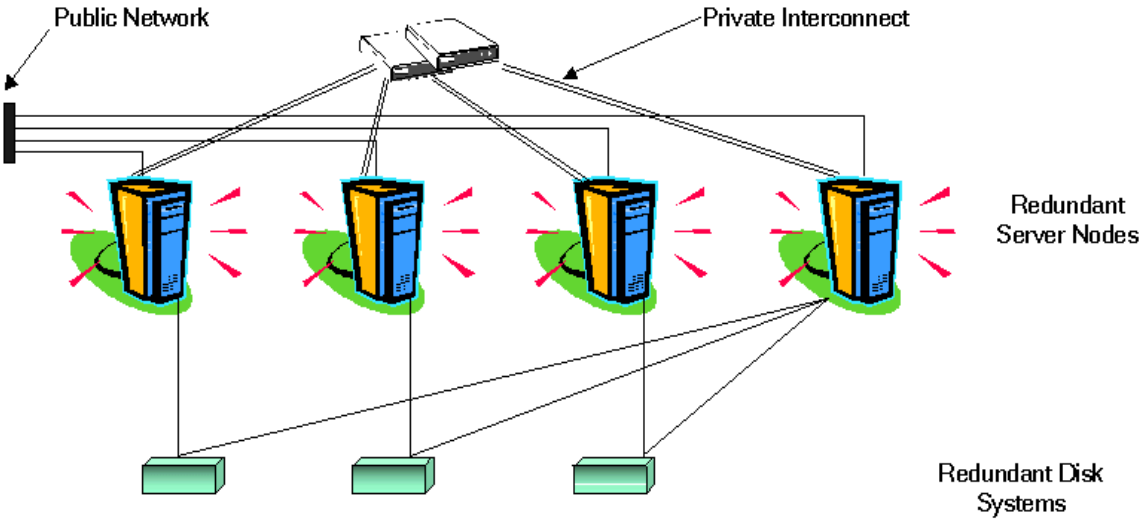
- Supports the use of a virtual IP address.
- Supports the use of a shared disk.
- Meets all of the requirements for NNMi as described in the *HP NNMi Software System and Device Support Matrix*.
- Includes the following additional patches:
  - *Windows*:
    - Microsoft hotfix for *Connecting to SMB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

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

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

Cluster architecture provides a single, globally coherent process and resource management view for the multiple nodes of a cluster. Figure 2 shows an example of a cluster architecture.

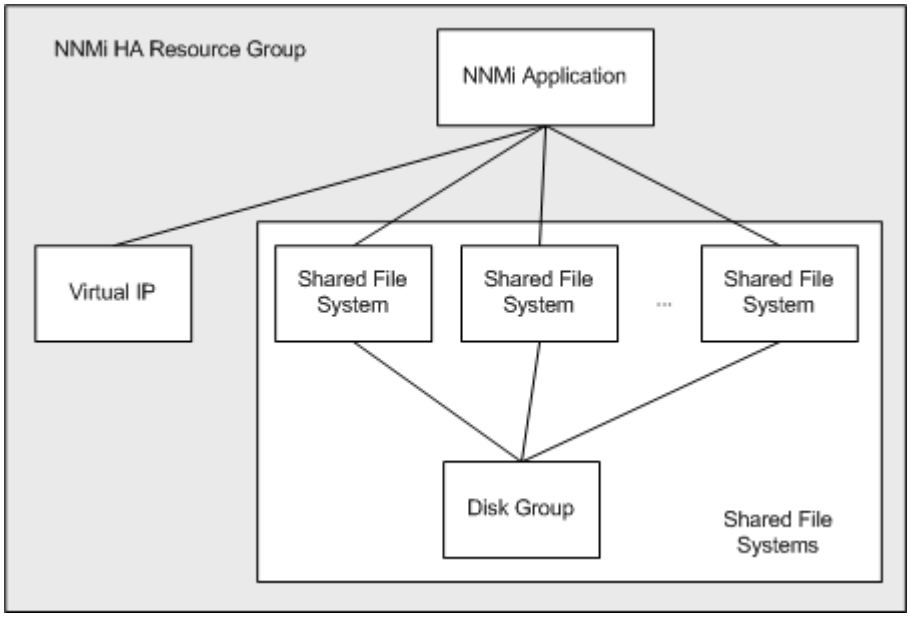
**Figure 2 Architecture of a High Availability Cluster**




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

In modern cluster environments such as HP ServiceGuard, Veritas Cluster Server, or Microsoft Cluster Services, applications are represented as compounds of resources, which are simple operations that enable applications to run in a cluster environment. The resources construct an **HA resource group**, which represents an application running in a cluster environment. Figure 3 shows an example of how an HA resource group may be organized.

**Figure 3 Typical HA Resource Group Layout**



This document uses the term *HA resource group* to designate a set of resources in any cluster environment. Each HA product uses a different name for the HA resource group. Table 3 lists the term for each supported HA product that equates to *HA resource group* for this document. (For the specific versions that are supported for each HA product, see Supported HA Products on page 78.)

**Table 3 Terminology for HA Resource Group in the Supported HA Products**

| HA Product | Abbreviation | Equivalent Term for HA Resource Group |
|---|---|---|
| HP ServiceGuard | SG | Package |
| Veritas Cluster Server | VCS | Service Group |
| Microsoft Cluster Services | MSCS | Resource Group |

## HA Terms

Table 4 lists and defines some common HA terms.

**Table 4 Common HA Terms**

| Term | Description |
|---|---|
| HA resource group | An application running in a cluster environment (under an HA product). An HA resource group can simultaneously be a cluster object that represents an application in a cluster. |
| Volume group | One or more disk drives that are configured to form a single large storage area. |
| Logical volume | An arbitrary-size space in a volume group that can be used as a separate file system or as a device swap space. |
| Primary cluster node | The first system on which the software product is installed, *and* the first system on which HA is configured.<br>The shared disk is mounted on the primary cluster node for initial set up.<br>The primary cluster node generally becomes the first active cluster node, but you do not need to maintain the primary designation after HA configuration is complete. The next time you update the HA configuration, another node might become the primary cluster node. |
| Secondary cluster node | Any system that is added to the HA configuration after the primary cluster node has been fully configured for HA. |
| Active cluster node | The system that is currently running the HA resource group. |
| Passive cluster node | Any system that is configured for HA but is not currently running the HA resource group. If the active cluster node fails, the HA resource group fails over to one of the available passive cluster nodes, which then becomes the active cluster node for that HA resource group. |

## NNMi HA Cluster Scenarios

For NNMi HA configuration, NNMi is installed on each system that will become part of an HA resource group. The NNMi database is located on a separate disk that is accessed by the NNMi programs running on each system. (Only one system, the active cluster node, accesses the shared disk at any given time.)

This approach is valid for the embedded and third-party database solutions.

Run the NNMi database backup and restore scripts on the active cluster node only.

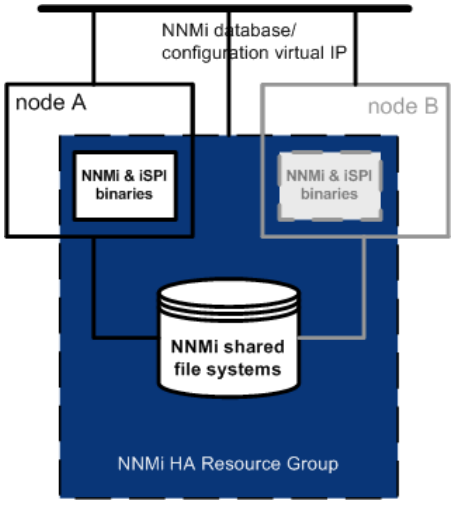
### NNMi-only scenario

Figure 4 shows a graphical representation of the NNMi HA cluster scenario. In this figure the NNMi HA resource group is synonymous with the NNMi HA cluster.

Node A and node B are each a fully installed NNMi management server that contains the NNMi program and any iSPIs that run on that system. The active cluster node accesses the shared disk for runtime data. Other products connect to NNMi by means of the virtual IP address of the HA resource group.

If the cluster contains more than two NNMi nodes, additional nodes are configured similarly to node B in Figure 4.

**Figure 4 Basic Scenario for NNMi HA Cluster**

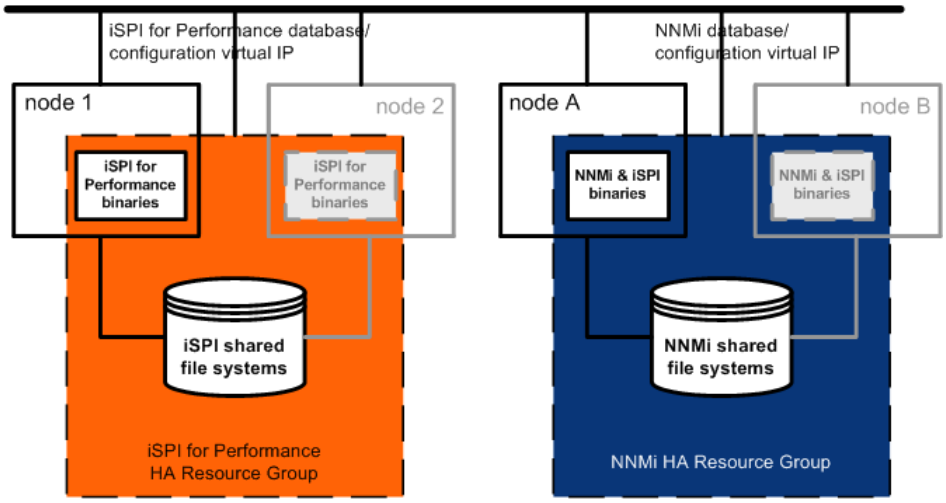



For information about how to implement this scenario, see Configuring NNMi for HA on page 84 and Configuring iSPIs for HA on page 89.

### NNMi and NNM iSPI for Performance on a dedicated server scenario

If you are running the NNM iSPI for Performance on a dedicated server, you can configure this iSPI to run as a separate HA resource group within the NNMi HA cluster, as shown in Figure 5.

**Figure 5 HA for NNMi and NNM iSPI for Performance on a Dedicated Server**



For information about how to implement this scenario, see Configuring NNMi for HA on page 84, Configuring iSPIs for HA on page 89, and Configuring the NNM iSPI for Performance on a Dedicated Server for HA on page 91.

Other options for the NNM iSPI for Performance on a dedicated server are as follows:

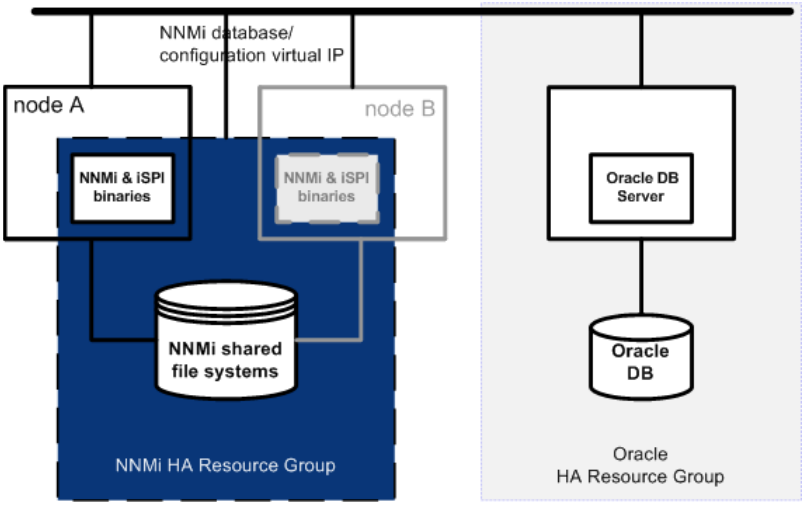
- Run the NNM iSPI for Performance on a single system with no HA. Use this approach while evaluating the iSPI and for environments where it is not critical for performance data to be always available.
- Configure the NNM iSPI for Performance to run under a different HA cluster than that for NNMi. In this case, you must manage the NNM iSPI for Performance's dependency on NNMi manually.

### NNMi with an Oracle database scenario

If your NNMi implementation utilizes Oracle for the main NNMi database, the Oracle database should be on a separate server for performance reasons, as shown in Figure 6. Therefore, you must configure two HA resource groups within the NNMi HA cluster:

- The NNMi resource group includes the NNMi nodes and a shared disk for NNMi data that is not stored in the Oracle database.
- The Oracle resource group contains the Oracle database server and the database disk.

**Figure 6 HA for NNMi with an Oracle Database**

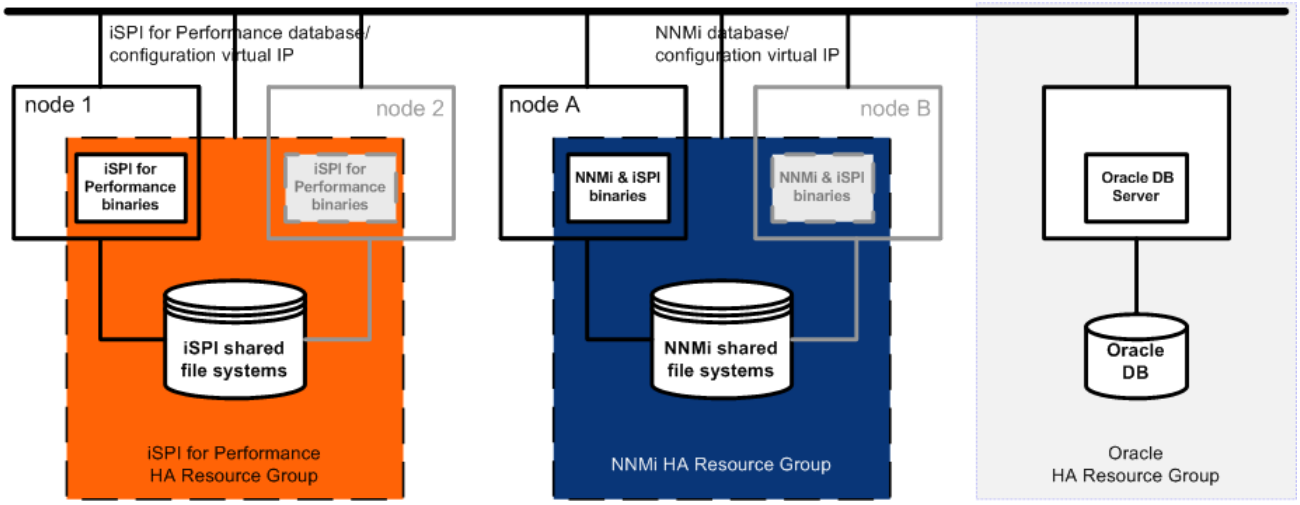



For information about how to implement this scenario, see Configuring NNMi for HA in an Oracle Environment on page 95 and Configuring iSPIs for HA on page 89.

### NNMi with an Oracle database and NNM iSPI for Performance on a dedicated server scenario

If your NNMi implementation utilizes Oracle for the main NNMi database and you are running the NNM iSPI for Performance on a dedicated server, you can configure three HA resource groups within the NNMi HA cluster, as shown in Figure 7.

**Figure 7 HA for NNMi with an Oracle Database and NNM iSPI for Performance on a Dedicated Server**




For information about how to implement this scenario, see Configuring NNMi for HA in an Oracle Environment on page 95, Configuring iSPIs for HA on page 89, and Configuring the NNM iSPI for Performance on a Dedicated Server for HA on page 91.

## Manpages

With regard to HA configuration, the NNMi manpages contain the following topics:

- nnm-ha
- nnmhaconfigure.ovpl
- nnmhaunconfigure.ovpl
- nnmhadisk.ovpl
- nnmhaclusterinfo.ovpl
- nnmhastartrg.ovpl
- nnmhastoprg.ovpl

On the Windows operating system, these manpages are available as text files.

# Configuring HA

## Configuring NNMi for HA

The two distinct phases of configuring NNMi for HA are as follows:

1. Copy the NNMi data files to the shared disk.
   - Do this task on the primary node, as described in step 2 through step 11 of Configuring NNMi on the Primary Cluster Node on page 86.
2. Configure NNMi to run under HA.
   - Do this task on the primary node, as described in step 12 through step 15 of Configuring NNMi on the Primary Cluster Node on page 86.
   - Do this task on the secondary node, as described in Configuring NNMi on the Secondary Cluster Nodes on page 88.

Designate one HA cluster node as the primary NNMi management server. This is the node that you expect to be active most of the time. Configure the primary node; then configure all other nodes in the HA cluster as secondary nodes.

You *cannot* configure NNMi for HA simultaneously on multiple cluster nodes. After the HA configuration process is completed on one cluster node, proceed with the HA configuration on the next node, and so forth until NNMi is configured for HA on all nodes in the cluster environment.

### NNMi HA Configuration Information

The HA configuration script collects information about the NNMi HA resource group. Table 5 lists the information that you will need for configuring the primary node. Gather this information before you begin the configuration procedure.

**Table 5 NNMi HA Primary Node Configuration Information**

| HA Configuration Item | Description |
|---|---|
| HA resource group | The name of the resource group for the HA cluster that contains NNMi.<br>For example: nnmtest1 |
| Virtual host short name | The short name for the virtual host. This hostname must map to the virtual IP address for the HA resource group. The `nslookup` command must be able to resolve the virtual host short name and the virtual IP address.<br>**NOTE:** If NNMi is unable to resolve the virtual host short name or the virtual host IP address, the HA configuration script could leave the system in an unstable state. Therefore, HP recommends that you implement a secondary naming strategy (such as entering the information in the `%SystemRoot%\system32\drivers\etc\hosts` file on the Windows operating system or `/etc/hosts` file on UNIX operating systems) in case DNS is not available during NNMi HA configuration. |
| Virtual host netmask | The subnet mask that is used with the virtual host IP address. |

**Table 5 NNMi HA Primary Node Configuration Information**

| HA Configuration Item | Description |
|---|---|
| Virtual host network interface | The network interface on which the virtual host IP address is running. For example:<br>• *Windows*: Local Area Connection<br>• *HP-UX*: lan0<br>• *Linux*: eth0<br>• *Solaris*: bge0 |
| Shared file system type | The type of shared disk configuration being used for the HA resource group. Possible values are:<br>• `disk`—The shared disk is a physically attached disk that uses a standard file system type. The HA configuration script can configure the shared disk. For more information, see the File system type entry in this table.<br>• `none`—The shared disk uses a configuration other than that described for the `disk` option, such as SAN or NFS. After running the HA configuration script, configure the shared disk as described in Configuring the Shared Disk on page 97. |
| File system type | The file system type of the shared disk (if the shared file system type is `disk`). The HA configuration scripts pass this value to the HA product so that it can determine how to validate the disk.<br>HP has tested the following shared disk formats:<br>• *Windows*: Basic (see A Note about Shared Disk Configuration on Windows 2003 on page 98)<br>• *HP-UX*: vxfs<br>• *Linux*: lvm2<br>• *Solaris*: vxfs<br>HP expects that other formats for a physically attached disk work for this HA implementation, and HP provides limited support in analyzing NNMi HA systems with other physically attached disks that use a standard file system type.<br>**NOTE**: If you use a shared disk format that HP has not tested, review the resource group configuration to verify that the values inserted by the NNMi HA configuration scripts are correct. |
| Disk group | (UNIX only) The name of the disk group for the NNMi shared file system. This name is based on the name of the HA resource group.<br>For example: nnmtest1-dg |
| Volume group | (UNIX only) The name of the volume group for the NNMi shared file system. This name is based on the name of the HA resource group.<br>For example: nnmtest1-vol |
| Mount point | The directory location for mounting the NNMi shared disk. This mount point must be consistent between systems. (That is, each node must use the same name for the mount point.) For example:<br>• *Windows*: `S:\`<br>• *UNIX*: `/nnmmount` |

## Configuring NNMi on the Primary Cluster Node

Complete the following procedure on the primary cluster node.

1. Verify that the system meets all of the requirements specified in Prerequisites to Configuring NNMi for HA on page 78.
2. If you have not already done so, install NNMi; then verify that NNMi is working correctly.

   You do not need to install iSPIs at this time; however, any iSPIs that are already installed on the NNMi management server can remain in place.
3. Back up the NNMi configuration:

   ```
   nnmbackup.ovpl -scope config -target <directory>
   ```

   For more information about this command, see NNMi Backup and Restore Tools on page 123.
4. Back up the NNMi license file by copying the following file to another location:
   - *Windows*: `%AUTOPASS_HOME%\data\LicFile.txt`

     To determine the value of `%AUTOPASS_HOME%`, examine the system environment variables for the computer.
   - *UNIX*: `/var/opt/OV/HPOvLIC/LicFile.txt`
5. Define the disk device group (and logical volume), consisting of at least one shared disk for the NNMi HA resource group. For example:
   - *Windows*: Use Disk Management to configure the disk mount point and format the disk.
   - *HP-UX* and *Linux*:

     ```
     pvcreate -f /dev/dsk/<physical_disk_name>
     mkdir /dev/<disk_group>
     mknod c 64 0x00h0 /dev/<disk_group>/group
     vgcreate /dev/rdsk/<physical_disk_name> <disk_group_name>
     lvcreate -L <disk_size> -n <logical_volume> <disk_group>
     newfs /dev/<disk_group>/r<logical_volume>
     ```
   - *Solaris*: Use the Symantec Veritas Storage Foundation as described here:

     Use `vxdiskadm` to add and initialize the disk.

     Use `vxassist make` to allocate disks by space.

     ```
     mkfs -F vxfs /dev/vx/dsk/<disk_group>/<logical_volume_group>
     ```

   For UNIX operating systems, a reference web site is:
   **http://www.unixguide.net/unixguide.shtml**
6. Create the directory mount point for the shared disk (for example, `S:\` or `/nnmmount`):
   - *Windows*: Use Windows Explorer and Disk Management.
   - *UNIX*: Verify that the shared disk directory mount point has been created with `root` as the user, `sys` as the group, and the permissions set to `555`. For example:

     ```
     ls -l /nnmmount
     ```

7 Mount the shared disk. For example:

- *Windows*: Use Disk Management.
- *HP-UX*:

  `mount /dev/<disk_group>/<logical_volume> /nnmmount`

- *Linux*:

  `mount /dev/<disk_group>/<logical_volume_group> /nnmmount`

- *Solaris*:

  `vxfs: mount /dev/vx/dsk/<disk_group>/<volume_group> /nnmmount`

8 Stop NNMi:

```
ovstop -c
```

9 Copy the NNMi database to the shared disk:

- *Windows*:

```
%NnmInstallDir%\misc\nnm\ha\nnmhadisk.ovpl NNM
-to <HA_mount_point>
```

- *UNIX*:

```
$NnmInstallDir/misc/nnm/ha/nnmhadisk.ovpl NNM
-to <HA_mount_point>
```

To prevent database corruption, run this command (with the `-to` option) only one time. For information about alternatives, see Re-Enable NNMi for HA after All Cluster Nodes are Unconfigured on page 115.

10 Unmount the shared disk.

11 (UNIX only) Deactivate the disk group:

```
umount <HA_mount_point>
```

12 Verify that NNMi is not running:

```
ovstop -c
```

13 Configure the NNMi HA resource group:

- *Windows*:

```
%NnmInstallDir%\misc\nnm\ha\nnmhaconfigure.ovpl NNM
```

- *UNIX*:

```
$NnmInstallDir/misc/nnm/ha/nnmhaconfigure.ovpl NNM
```

Table 5 on page 84 describes the information that this command requests.

14 In step 13, what value did you specify for the shared file system type (as described for Shared file system type and File system type in Table 5 on page 84)?

- For type `disk`, the `nnmhaconfigure.ovpl` command configured the shared disk. Continue with step 15.
- For type `none`, configure the shared disk as described in Configuring the Shared Disk on page 97; then continue with step 15.

15 Start the NNMi HA resource group:

- *Windows*:

  **%NnmInstallDir%\misc\nnm\ha\nnmhastartrg.ovpl NNM *<resource_group>***

- *UNIX*:

  **$NnmInstallDir/misc/nnm/ha/nnmhastartrg.ovpl NNM *<resource_group>***

If NNMi does not start correctly, see Troubleshooting the HA Configuration on page 113.

Now that NNMi is running under HA, *do not* use the `ovstart` and `ovstop` commands for normal operation. Use these commands only when instructed to do so for HA maintenance purposes.

## Configuring NNMi on the Secondary Cluster Nodes

Complete the following procedure on one secondary cluster node at a time.

1 If you have not already done so, complete the procedure for Configuring NNMi on the Primary Cluster Node on page 86.

2 Verify that the system meets all of the requirements specified in Prerequisites to Configuring NNMi for HA on page 78.

3 If you have not already done so, install NNMi, and then verify that NNMi is working correctly.

You do not need to install iSPIs at this time; however, any iSPIs that are already installed on the NNMi management server can remain in place.

4 Create a mount point for the shared disk (for example, `S:\` or `/nnmmount`).

This mount point must use the same name as the mount point that you created in step 6 of the procedure Configuring NNMi on the Primary Cluster Node.

5 Configure the NNMi HA resource group:

- *Windows*: **%NnmInstallDir%\misc\nnm\ha\nnmhaconfigure.ovpl NNM**
- *UNIX*: **$NnmInstallDir/misc/nnm/ha/nnmhaconfigure.ovpl NNM**

Supply the HA resource group name when the command requests this information.

6 Verify that the configuration was successful:

- *Windows*:

  **%NnmInstallDir%\misc\nnm\ha\nnmhaclusterinfo.ovpl -group *<resource_group>* -nodes**

- *UNIX*:

  **$NnmInstallDir/misc/nnm/ha/nnmhaclusterinfo.ovpl -group *<resource_group>* -nodes**

The command output lists all nodes that have been configured for the specified HA resource group.

7 Optionally, test the configuration by taking the resource group on the primary node offline and then bringing the resource group on the secondary node online.

# Configuring iSPIs for HA

## iSPI HA Configuration Information

As of NNMi version 8.10, the NNMi-supplied HA configuration commands can be used to configure iSPIs to run in the NNMi HA resource group. Each iSPI must meet the following requirements:

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

If you are running the NNM iSPI for Performance on a separate system, see Configuring the NNM iSPI for Performance on a Dedicated Server for HA on page 91.

This configuration process requires the base name of the Perl module that the iSPI installs on the NNMi management server. Table 6 lists the names that were known when this chapter was last updated. To determine what value to use for other iSPIs, see the appropriate iSPI chapter in the Integrations and Plug-ins section.

**Table 6 iSPI Perl Module Names**

| iSPI | Perl Module Base Name |
|---|---|
| NNM iSPI for Performance | `PerfSPIHA` |

iSPI configuration for HA is order independent. If you plan to run multiple iSPIs on the NNMi management server, you can use either of the following process flows:

- Complete both step 3 and step 4 of the Configuring an Add-On iSPI (All Other iSPIs) procedure for the first iSPI, and then complete both step 3 and step 4 for the second iSPI, and so forth.
- Complete step 3 of the Configuring an Add-On iSPI (All Other iSPIs) procedure for all iSPIs, and then complete step 4 for all iSPIs on one passive cluster node at a time.

## Configuring the NNM iSPI for Performance as an Add-on iSPI

The simplest procedure for configuring same-system NNM iSPI for Performance to run under the NNMi HA resource group is as follows:

1. Completely configure and start the NNMi HA resource group for all nodes in the NNMi HA cluster as described in Configuring NNMi for HA on page 84 or Configuring NNMi for HA in an Oracle Environment on page 95. Verify that the cluster is properly configured as described in step 6 on page 88.

The tools that configure iSPIs to run under HA interact with NNMi during the configuration process. For the simplest configuration experience, ensure that all nodes in the NNMi HA cluster are up during the entire iSPI configuration process.

2. On the active node, verify that the NNMi services are running:

   ```
   ovstatus -c
   ```

   All NNMi services should show the state RUNNING.

3 On the active node in the NNMi HA cluster, do the following:

   a Run the NNM iSPI for Performance enablement script.

   b Install the NNM iSPI for Performance.

   For information about how to complete these steps, see the *NNM iSPI for Performance Installation Guide*.

4 On each passive node in the NNMi HA cluster, do the following:

   a Run the NNM iSPI for Performance enablement script.

   b Install the NNM iSPI for Performance.

## Configuring an Add-On iSPI (All Other iSPIs)

Best practice

The NNM iSPI for Performance installation process automatically configures the add-on iSPI to run under the NNMi HA resource group. If you have not yet installed the NNM iSPI for Performance, see Configuring the NNM iSPI for Performance as an Add-on iSPI.

To configure an add-on iSPI to run under the HA resource group, follow these steps:

1 Completely configure and start the NNMi HA resource group for all nodes in the NNMi HA cluster as described in Configuring NNMi for HA on page 84 or Configuring NNMi for HA in an Oracle Environment on page 95. Verify that the cluster is properly configured as described in step 6 on page 88.

The tools that configure iSPIs to run under HA interact with NNMi during the configuration process. For the simplest configuration experience, ensure that all nodes in the NNMi HA cluster are up during the entire iSPI configuration process.

2 On the active node in the NNMi HA cluster, verify that the NNMi services are running:

```
ovstatus -c
```

   All NNMi services should show the state RUNNING.

3 On the active node in the NNMi HA cluster, configure the iSPI for HA:

   a If you have not already done so, install the iSPI.

   b Add the iSPI to the NNMi HA resource group:

      — *Windows*:

```
%NnmInstallDir%\misc\nnm\ha\nnmhaconfigure.ovpl NNM
-addon <iSPI_PM_Name>
```

      — *UNIX*:

```
$NnmInstallDir/misc/nnm/ha/nnmhaconfigure.ovpl NNM
-addon <iSPI_PM_Name>
```

   Where *<iSPI_PM_Name>* is the base name of the Perl module that the iSPI installs on the NNMi management server. To determine what value to use for *<iSPI_PM_Name>*, see Table 6 on page 89 or the appropriate iSPI chapter in the Integrations and Plug-ins section.

4 Configure the iSPI for HA on each passive node in the NNMi HA cluster:

   a If you have not already done so, install the iSPI.

   b Add the iSPI to the NNMi HA resource group:

      — *Windows*:

```
%NnmInstallDir%\misc\nnm\ha\nnmhaconfigure.ovpl NNM
-addon <iSPI_PM_Name>
```

      — *UNIX*:

```
$NnmInstallDir/misc/nnm/ha/nnmhaconfigure.ovpl NNM
-addon <iSPI_PM_Name>
```

5 Verify the configuration:

   - *Windows*:

```
%NnmInstallDir%\misc\nnm\ha\nnmhaclusterinfo.ovpl
-config NNM -get NNM_ADD_ON_PRODUCTS
```

   - *UNIX*:

```
$NnmInstallDir/misc/nnm/ha/nnmhaclusterinfo.ovpl
-config NNM -get NNM_ADD_ON_PRODUCTS
```

   The command output lists the add-on iSPI configurations in the format `<iSPI_PM_Name>[hostname_list]`. For example:

```
PerfSPIHA[hostname1, hostname2]
```

## Configuring the NNM iSPI for Performance on a Dedicated Server for HA

### NNM iSPI for Performance HA Configuration Information

The HA configuration script collects information about the NNM iSPI for Performance HA resource group. Table 7 lists the information that you will need for configuring the primary node. Gather this information before you begin the configuration procedure.

**Table 7 NNM iSPI for Performance HA Primary Node Configuration Information**

| HA Configuration Item | Description |
|---|---|
| HA resource group | The name of the resource group for the HA cluster that contains the NNM iSPI for Performance.<br>For example: perftest1 |
| Virtual host short name | The short name for the virtual host. This hostname must map to the virtual IP address for the HA resource group. The `nslookup` command must be able to resolve the virtual host short name and the virtual IP address.<br>**NOTE:** If the NNMi is unable to resolve the virtual host short name or the virtual host IP address, the HA configuration script could leave the system in an unstable state. Therefore, HP recommends that you implement a secondary naming strategy (such as entering the information in the `%SystemRoot%\system32\drivers\etc\hosts` file on the Windows operating system or `/etc/hosts` file on UNIX operating systems) in case DNS is not available during NNM iSPI for Performance HA configuration. |
| Virtual host netmask | The subnet mask that is used with the virtual host IP address. |
| Virtual host network interface | The network interface on which the virtual host IP address is running. For example:<br>• *Windows*: Local Area Connection<br>• *HP-UX*: lan0<br>• *Linux*: eth0<br>• *Solaris*: bge0 |
| Shared file system type | The type of shared disk configuration being used for the HA resource group. Possible values are:<br>• `disk`—The shared disk is a physically attached disk that uses a standard file system type. The HA configuration script can configure the shared disk. For more information, see the File system type entry in this table.<br>• `none`—The shared disk uses a configuration other than that described for the `disk` option, such as SAN or NFS. After running the HA configuration script, configure the shared disk as described in Configuring the Shared Disk on page 97. |

**Table 7 NNM iSPI for Performance HA Primary Node Configuration Information**

| HA Configuration Item | Description |
|---|---|
| File system type | The file system type of the shared disk (if the shared file system type is `disk`). The HA configuration scripts pass this value to the HA product so that it can determine how to validate the disk.<br>HP has tested the following shared disk formats:<br>• *Windows*: Basic (see A Note about Shared Disk Configuration on Windows 2003 on page 98)<br>• *HP-UX*: vxfs<br>• *Linux*: lvm2<br>• *Solaris*: vxfs<br>HP expects that other formats for a physically attached disk work for this HA implementation, and HP provides limited support in analyzing NNMi HA systems with other physically attached disks that use a standard file system type.<br>**NOTE**: If you use a shared disk format that HP has not tested, review the resource group configuration to verify that the values inserted by the NNMi HA configuration scripts are correct. |
| Disk group | (UNIX only) The name of the disk group for the NNM iSPI for Performance shared file system. This name is based on the name of the HA resource group.<br>For example: perftest1-dg |
| Volume group | (UNIX only) The name of the volume group for the NNM iSPI for Performance shared file system. This name is based on the name of the HA resource group.<br>For example: perftest1-vol |
| Mount point | The directory location for mounting the NNM iSPI for Performance shared disk. This mount point must be consistent between systems. (That is, each node must use the same name for the mount point.) For example:<br>• *Windows*: `P:\`<br>• *UNIX*: `/perfmount` |

## Configuring the NNM iSPI for Performance on the Primary Cluster Node

Best practice

If you have not already done so, run the NNM iSPI for Performance enablement script on each node in the NNMi HA cluster.

Complete the following procedure on the primary cluster node:

1. If you have not already done so, install the NNM iSPI for Performance. In the **NNM iSPI for Performance Configuration** window, enter the required details, and *do not* start the service.

   For information about the fields in this window, see the *NNM iSPI for Performance Installation Guide*.

2. Define the disk device group (and logical volume), consisting of at least one shared disk for the performance HA resource group. For examples, see step 5 on page 86.

3. Create the directory mount point for the shared disk (for example, `P:\` or `/perfmount`). For examples, see step 6 on page 86.

4 Mount the shared disk. For examples, see step 7 on page 87.

5 Verify that the NNM iSPI for Performance is not running:

- *Windows*:

```
%NnmInstallDir%\NNMPerformanceSPI\bin\statusSPI.ovpl
%NnmInstallDir%\NNMPerformanceSPI\bin\statusERS.ovpl
```

- *UNIX*:

```
$NnmInstallDir/NNMPerformanceSPI/bin/statusSPI.ovpl
$NnmInstallDir/NNMPerformanceSPI/bin/statusERS.ovpl
```

6 Configure the performance HA resource group:

- *Windows*:

```
%NnmInstallDir%\misc\nnm\ha\nnmhaconfigure.ovpl PerfSPIHA
```

- *UNIX*:

```
$NnmInstallDir/misc/nnm/ha/nnmhaconfigure.ovpl PerfSPIHA
```

Table 7 on page 92 describes the information that this command requests.

7 In step 6, what value did you specify for the shared file system type (as described for Shared file system type and File system type in Table 7 on page 92)?

- For type `disk`, the `nnmhaconfigure.ovpl` command configured the shared disk. Continue with step 8.
- For type `none`, configure the shared disk as described in Configuring the Shared Disk on page 97; then continue with step 8.

8 Start the performance HA resource group:

- *Windows*:

```
%NnmInstallDir%\misc\nnm\ha\nnmhastartrg.ovpl PerfSPIHA
<resource_group>
```

- *UNIX*:

```
$NnmInstallDir/misc/nnm/ha/nnmhastartrg.ovpl PerfSPIHA
<resource_group>
```

If the NNM iSPI for Performance does not start correctly, see General HA Troubleshooting on page 113.

## Configuring the NNM iSPI for Performance on the Secondary Cluster Nodes

Complete the following procedure on one secondary cluster node at a time:

1 If you have not already done so, complete the procedure for Configuring the NNM iSPI for Performance on the Primary Cluster Node on page 93.

2 Create a mount point for the shared disk (for example, `P:\` or `/perfmount`).

This mount point must use the same name as the mount point that you created in step 2 of the procedure Configuring the NNM iSPI for Performance on the Primary Cluster Node.

3 Configure the NNM iSPI for Performance HA resource group:

- *Windows*:

```
%NnmInstallDir%\misc\nnm\ha\nnmhaconfigure.ovpl PerfSPIHA
```

- *UNIX*:

```
$NnmInstallDir/misc/nnm/ha/nnmhaconfigure.ovpl PerfSPIHA
```

Supply the performance HA resource group name when the command requests this information.

4 Verify that the configuration was successful:

- *Windows*:

```
%NnmInstallDir%\misc\nnm\ha\nnmhaclusterinfo.ovpl
-group <resource_group> -nodes
```

- *UNIX*:

```
$NnmInstallDir/misc/nnm/ha/nnmhaclusterinfo.ovpl
-group <resource_group> -nodes
```

The command output lists all nodes that have been configured for the specified HA resource group.

5 Optionally, test the configuration by taking the resource group on the primary node offline and then bringing the resource group on the secondary node online.

## Configuring NNMi for HA in an Oracle Environment

This sections presents a high-level overview of the process for configuring NNMi with an Oracle database to run under HA. The number of possible Oracle configurations is large, and the configuration process can vary according to the Oracle release. For the most accurate information about configuring Oracle to run under HA and creating an NNMi dependency on the Oracle HA resource group, see the HA product documentation for information about how that product supports an Oracle database. You can also go to the Oracle web site (**http://www.oracle.com**) for information on the appropriate Oracle configuration for your HA product.

### NNMi with Oracle HA Configuration Information

When Oracle and NNMi both run under HA, each product must be in a separate HA resource group. The Oracle resource group must be fully started before the NNMi resource group starts. If both resource groups are in the same HA cluster, you can modify the cluster configuration to set resource group ordering. If the resource groups are in different HA clusters, ensure that the NNMi resource group dependency on the Oracle resource group is met.

The NNMi resource group must include a shared disk for the NNMi data that is not stored in the Oracle database.

### Configuring NNMi with Oracle for HA

1 If you plan to run Oracle under HA, complete that configuration first.

2 On the primary NNMi node, install NNMi. During installation, use the virtual IP address or hostname for the Oracle HA resource group (if applicable).

3. On the primary NNMi node, configure NNMi to run under HA as described in Configuring NNMi on the Primary Cluster Node on page 86.
4. Set up the NNMi dependency on the Oracle HA resource group.

   For specific instructions, see the HA product documentation.
5. On the secondary NNMi node, install NNMi. During installation, use the virtual IP address or hostname for the Oracle HA resource group (if applicable).
6. On the secondary NNMi node, configure NNMi to run under HA described in Configuring NNMi on the Secondary Cluster Nodes on page 88.
7. For each additional secondary NNMi node, repeat step 5 and step 6.

# Shared NNMi Data

This implementation of NNMi running under HA requires the use of a separate disk for sharing files between all NNMi nodes in the HA cluster.

NNMi implementations that use Oracle as the primary database also require the use of a separate disk for shared data.

## Data on the NNMi Shared Disk

This section lists the NNMi data files that are maintained on the shared disk when NNMi is running under HA.

The locations are mapped to the shared disk location as follows:

- *Windows*:
  - `%NnmInstallDir%` maps to `%HA_MOUNT_POINT%\NNM\installDir`
  - `%NnmDataDir%` maps to `%HA_MOUNT_POINT%\NNM\dataDir`
- *UNIX*:
  - `$NnmInstallDir` maps to `$HA_MOUNT_POINT/NNM/installDir`
  - `$NnmDataDir` maps to `$HA_MOUNT_POINT/NNM/dataDir`

The directories that are moved to the shared disk are as follows:

- *Windows*:
  - `%NnmDataDir%\shared\nnm\databases\Postgres`
    The embedded database; present for all implementations.
  - `%NnmDataDir%\log\nnm`
    The NNMi logging directory.
  - `%NnmDataDir%\shared\nnm\databases\eventdb`
    The pmd events database.
  - `%NnmInstallDir%\nonOV\jboss\nms\server\nms\data`
    The transactional store used by ovjboss.

- *UNIX*:
  - `$NnmDataDir/shared/nnm/databases/Postgres`
    The embedded database; present for all implementations.
  - `$NnmDataDir/log/nnm`
    The NNMi logging directory.
  - `$NnmDataDir/shared/nnm/databases/eventdb`
    The pmd events database.
  - `$NnmInstallDir/nonOV/jboss/nms/server/nms/data`
    The transactional store used by ovjboss.

The `nnmhadisk.ovpl` command copies these files to and from the shared disk. Run this command as the instructions in this chapter indicate. For a summary of the command syntax, see the *nnm-ha* manpage.

## Configuration File Replication

The NNMi HA implementation uses file replication to maintain copies of the NNMi configuration files on all NNMi nodes in the HA cluster. By default, the NNMi command `nnmdatareplicator.ovpl` manages file replication. This command copies the NNMi configuration files from the active node to a passive node during the failover process. The `nnmdatareplicator.conf` file specifies the NNMi folders and files that are included in data replication.

For information about the data replication process, see the *nnm-ha* manpage.

## Configuring the Shared Disk

If the shared disk is of a format that HP has tested (as listed in Table 5 on page 84), the HA configuration script prepares the shared disk, and you can safely ignore this section.

If the shared disk uses a non-tested configuration, such as SAN or NFS, you must prepare the disk manually. Enter the value `none` for the file system type during HA configuration, and then modify the disk configuration file according to the product documentation. For example:

- *HP-UX*:

  `/etc/cmcluster/<resource_group>/<resource_group>.cntl`

- *Linux*:

  `/usr/local/cmcluster/conf/<resource_group>/<resource_group>.cntl`

- *Solaris*: Add disk entries and links to the HA configuration file by using the `/opt/VRTSvcs/bin/hares` command.
- *Windows*: Use Cluster Administrator (`cluadmin.exe`) or the `cluster.exe` command to add resources to the resource group.

### A Note about Shared Disk Configuration on Windows 2003

According to Microsoft Knowledge Base article 237853, dynamic disks are not supported in Windows 2003 with Microsoft Cluster Services. To ensure the correct disk configuration, review the information on the following web sites:

- **http://support.microsoft.com/kb/237853**
- **http://www.petri.co.il/ difference_between_basic_and_dynamic_disks_in_windows_xp_2000_2003.htm**

## Licensing NNMi in an HA Cluster

For NNMi running under HA, the NNMi license is tied to the virtual IP address of the cluster. Therefore, an HA cluster requires only one NNMi license.

To correctly install the NNMi license for an HA cluster, perform these steps on the active NNMi cluster node:

1. At a command prompt, enter the following command:
   - *Windows:* **%NnmInstallDir%\bin\nnmlicense.ovpl NNM -g**
   - *UNIX*: **$NnmInstallDir/bin/nnmlicense.ovpl NNM -g**
2. In the **License Password** dialog box, click **Request License**.
3. Follow the instructions on the screen to obtain the permanent license password for the virtual IP address of the HA cluster.
4. At a command prompt, enter the following command to update your system and to store your license data files:
   - *Windows*:

     **%NnmInstallDir%\bin\nnmlicense.ovpl NNM -f *<license_file>***
   - *UNIX*:

     **$NnmInstallDir/bin/nnmlicense.ovpl NNM -f *<license_file>***
5. Update the file `licenses.txt` in the `NNM` directory on the shared disk (for example `S:\NNM\licenses.txt` or `/nnmount/NNM/licenses.txt`) with the new information from the license file on the active node, which is in the following location:
   - *Windows*: `%AUTOPASS_HOME%\data\LicFile.txt`

     To determine the value of `%AUTOPASS_HOME%`, examine the system environment variables for the computer.
   - *UNIX*: `/var/opt/OV/HPOvLIC/LicFile.txt`

   Do one of the following:
   - If this file exists on the shared disk, append the new license keys in `LicFile.txt` on the active node to `licenses.txt` on the shared disk.
   - If this file does not exist on the shared disk, copy `LicFile.txt` from the active node to `licenses.txt` in the `NNM` directory on the shared disk.

# Maintaining the HA Configuration

## Maintenance Mode

### Putting an HA Resource Group into Maintenance Mode

Putting an HA resource group into maintenance mode disables HA resource group monitoring. When an HA resource group is in maintenance mode, stopping and starting the products in that HA resource group do not cause failover.

To put an HA resource group into maintenance mode, on the active cluster node, create the following file:

- *Windows*: `%NnmDataDir%\hacluster\<resource_group>\maintenance`
- *UNIX*: `$NnmDataDir/hacluster/<resource_group>/maintenance`

  The file can be empty.

In NNMi 8.0x, the maintenance file is:

- *Windows*: `%NnmDataDir%\hacluster\maint_NNM`
- *UNIX*: `$NnmDataDir/hacluster/<resource_group>/maint_NNM`

### Removing an HA Resource Group from Maintenance Mode

Taking an HA resource group out of maintenance mode re-enables HA resource group monitoring. Stopping the products in that HA resource group causes the HA resource group to fail over to a passive cluster node.

To remove an HA resource group from maintenance mode, delete the maintenance file from the node that was the active cluster node before maintenance was initiated. This file is described in Putting an HA Resource Group into Maintenance Mode.

## Maintaining NNMi in an HA Cluster

### Starting and Stopping NNMi

While NNMi is running under HA, *do not* use the `ovstart` and `ovstop` commands unless instructed to do so for HA maintenance purposes. For normal operation, use the NNMi-provided HA commands or the appropriate HA product commands for starting and stopping resource groups.

### Changing NNMi Hostnames and IP Addresses in a Cluster Environment

A node in a cluster environment can have more than one IP address and hostname. If a node becomes a member of another subnet, you might need to change its IP addresses. As a result, the IP address or fully-qualified domain name might change.

For example, on UNIX systems, the IP address and the related hostname are generally configured in one of the following:

- /etc/hosts
- Domain Name Service (DNS)
- Network Information Service (NIS on HP-UX or Linux, NIS+ on Solaris)

NNMi also configures the hostname and IP address of the management server for the managed node in the NNMi database.

If you are moving from a non-name-server environment to a name-server environment (that is, DNS or BIND), make sure that the name server can resolve the new IP address.

Hostnames work within IP networks to identify a managed node. While a node may have many IP addresses, the hostname is used to pinpoint a specific node. The system hostname is the string returned when you use the `hostname` command.

To change the virtual hostname or IP address of the management server, perform these steps on the active NNMi cluster node:

1. At a command prompt, enter the following command:
   - *Windows:* **`%NnmInstallDir%\bin\nnmlicense.ovpl NNM -g`**
   - *UNIX*: **`$NnmInstallDir/bin/nnmlicense.ovpl NNM -g`**
2. In the **License Password** dialog box, click **Request License**.
3. Follow the instructions on the screen to obtain the permanent license password for the new virtual IP address of the HA cluster.
4. At a command prompt, enter the following command to update your system and to store your license data files:
   - *Windows*:

     **`%NnmInstallDir%\bin\nnmlicense.ovpl NNM -f <license_file>`**
   - *UNIX*:

     **`$NnmInstallDir/bin/nnmlicense.ovpl NNM -f <license_file>`**
5. Update the file `licenses.txt` in the `NNM` directory on the shared disk (for example `S:\NNM\licenses.txt` or `/nnmount/NNM/licenses.txt`) with the new information from the license file on the active node, which is in the following location:
   - *Windows*: `%AUTOPASS_HOME%\data\LicFile.txt`

     To determine the value of `%AUTOPASS_HOME%`, examine the system environment variables for the computer.
   - *UNIX*: `/var/opt/OV/HPOvLIC/LicFile.txt`

   Do one of the following:
   - If this file exists on the shared disk, append the new license keys in `LicFile.txt` on the active node to `licenses.txt` on the shared disk.
   - If this file does not exist on the shared disk, copy `LicFile.txt` from the active node to `licenses.txt` in the `NNM` directory on the shared disk.
6. Put the HA resource group into maintenance mode as described in Putting an HA Resource Group into Maintenance Mode on page 99.

7 Stop NNMi:

```
ovstop -c
```

8 Change the IP address or node name of the NNMi management server:

a In the `ov.conf` file, edit the `NNM_INTERFACE` entry to be the new hostname or IP address.

b In the `ovspmd.auth` file, edit any lines containing the old hostname to contain the new hostname.

The `ov.conf` and `ovspmd.auth` files are available in the following location:

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

9 Set the cluster configuration:

a Stop the NNMi HA resource group:

— *Windows*:

```
%NnmInstallDir%\misc\nnm\ha\nnmhastoprg.ovpl NNM
<resource_group>
```

— *UNIX*:

```
$NnmInstallDir/misc/nnm/ha/nnmhastoprg.ovpl NNM
<resource_group>
```

b Change the cluster configuration to use the new IP address:

— *Microsoft Cluster Services*:

In Cluster Administrator, open ***<resource_group>***.

Double-click ***<resource_group>*-ip**, select **Parameters**, and the enter the new IP address.

— *HP ServiceGuard*: On the active HA cluster node, edit the following file:

*HP-UX*: `/etc/cmcluster/<resource_group>/<resource_group>.cntl`

*Linux*: `/usr/local/cmcluster/conf/<resource_group>/<resource_group>.cntl`

Replace `IP[0]=<old_IP_address>` with `IP[0]=<new_IP_address>`. Then use `cmapplyconf` to update all other systems.

— *VERITAS Cluster Server*:

```
$NnmInstallDir/misc/nnm/ha/nnmhargconfigure.ovpl NNM
<resource_group> -set_value <resource_group>-ip
Address <new_IP_address>
```

c Start the NNMi HA resource group:

— *Windows*:

```
%NnmInstallDir%\misc\nnm\ha\nnmhastartrg.ovpl NNM
<resource_group>
```

— *UNIX*:

```
$NnmInstallDir/misc/nnm/ha/nnmhastartrg.ovpl NNM
<resource_group>
```

10 Verify that NNMi started correctly:

```
ovstatus -c
```

All NNMi services should show the state RUNNING.

11 Take the HA resource group out of maintenance mode as described in Removing an HA Resource Group from Maintenance Mode on page 99.

### Stopping NNMi Without Causing Failover

When you need to perform NNMi maintenance, you can stop NNMi on the active cluster node without causing failover to a currently passive node. Follow these steps on the active cluster node:

1 Put the HA resource group into maintenance mode as described in Putting an HA Resource Group into Maintenance Mode on page 99.

2 Stop NNMi:

```
ovstop -c
```

### Restarting NNMi after Maintenance

If you have stopped NNMi in the manner that prevents failover, follow these steps to restart NNMi and HA monitoring:

1 Start NNMi:

```
ovstart -c
```

2 Verify that NNMi started correctly:

```
ovstatus -c
```

All NNMi services should show the state RUNNING.

3 Take the HA resource group out of maintenance mode as described in Removing an HA Resource Group from Maintenance Mode on page 99.

## Maintaining Add-on iSPIs in an NNMi HA Cluster

The iSPIs are closely linked to NNMi. When add-on iSPIs are installed on the nodes in the NNMi HA cluster, use the NNMi HA cluster maintenance procedures as written.

## Maintaining the NNM iSPI for Performance on a Dedicated Server in an HA Cluster

### Starting and Stopping the NNM iSPI for Performance

While the NNM iSPI for Performance is running under HA, *do not* use the `startSPI`, `startERS`, `stopSPI`, and `stopERS` commands unless instructed to do so for HA maintenance purposes. Also, do *not* use the **Start** and **Stop** buttons in the NNM iSPI for Performance user interface. For normal operation, use the NNMi-provided HA commands or the appropriate HA product commands for starting and stopping resource groups.

## Changing the NNM iSPI for Performance System in a Cluster Environment

Follow the procedure for Changing NNMi Hostnames and IP Addresses in a Cluster Environment on page 99, making the appropriate adjustments for working with the NNM iSPI for Performance.

## Stopping NNM iSPI for Performance Without Causing Failover

When you need to perform NNM iSPI for Performance maintenance, you can stop NNM iSPI for Performance on the active node without causing failover to a currently passive node. Follow these steps on the active node:

1. Put the HA resource group into maintenance mode as described in Putting an HA Resource Group into Maintenance Mode on page 99.

   The file can be empty.

2. Stop the NNM iSPI for Performance:

   - *Windows*:

     ```
     %NnmInstallDir%\NNMPerformanceSPI\bin\stopSPI.ovpl
     %NnmInstallDir%\NNMPerformanceSPI\bin\stopERS.ovpl
     ```

   - *UNIX*:

     ```
     $NnmInstallDir/NNMPerformanceSPI/bin/stopSPI.ovpl
     $NnmInstallDir/NNMPerformanceSPI/bin/stopERS.ovpl
     ```

## Restarting NNM iSPI for Performance after Maintenance

If you have stopped NNM iSPI for Performance in the manner that prevents failover, follow these steps to restart NNM iSPI for Performance and HA monitoring:

1. Start the NNM iSPI for Performance:

   - *Windows*:

     ```
     %NnmInstallDir%\NNMPerformanceSPI\bin\startSPI.ovpl
     %NnmInstallDir%\NNMPerformanceSPI\bin\startERS.ovpl
     ```

   - *UNIX*:

     ```
     $NnmInstallDir/NNMPerformanceSPI/bin/startSPI.ovpl
     $NnmInstallDir/NNMPerformanceSPI/bin/startERS.ovpl
     ```

2. Verify that NNM iSPI for Performance started correctly:

   - *Windows*:

     ```
     %NnmInstallDir%\NNMPerformanceSPI\bin\statusSPI.ovpl
     %NnmInstallDir%\NNMPerformanceSPI\bin\statusERS.ovpl
     ```

   - *UNIX*:

     ```
     $NnmInstallDir/NNMPerformanceSPI/bin/statusSPI.ovpl
     $NnmInstallDir/NNMPerformanceSPI/bin/statusERS.ovpl
     ```

3. Take the HA resource group out of maintenance mode as described in Removing an HA Resource Group from Maintenance Mode on page 99.

# Unconfiguring HA

## Unconfiguring NNMi from an HA Cluster

The process of removing an NNMi node from an HA cluster involves undoing the HA configuration for that instance of NNMi. You can then run that instance of NNMi as a standalone management server, or you can uninstall NNMi from that node.

If you want to keep NNMi configured for high availability, the HA cluster must contain one node that is actively running NNMi and at least one passive NNMi node. If you want to completely remove NNMi from the HA cluster, unconfigure the HA functionality on all nodes in the cluster.

To completely unconfigure NNMi from an HA cluster, follow these steps:

1. Determine which node in the HA cluster is active:
   - *Windows*:

     ```
     %NnmInstallDir%\misc\nnm\ha\nnmhaclusterinfo.ovpl
     -group <resource_group> -state
     ```

   - *UNIX*:

     ```
     $NnmInstallDir/misc/nnm/ha/nnmhaclusterinfo.ovpl
     -group <resource_group> -state
     ```

2. On each passive node, unconfigure the add-on iSPIs from the HA cluster. For each iSPI, run the following command:
   - *Windows*:

     ```
     %NnmInstallDir%\misc\nnm\ha\nnmhaunconfigure.ovpl NNM
     -addon <iSPI_PM_Name>
     ```

   - *UNIX*:

     ```
     $NnmInstallDir/misc/nnm/ha/nnmhaunconfigure.ovpl NNM
     -addon <iSPI_PM_Name>
     ```

   Where *<iSPI_PM_Name>* is the base name of the Perl module that the iSPI installs on the NNMi management server. To determine what value to use for *<iSPI_PM_Name>*, see Table 6 on page 89 or the appropriate iSPI chapter in the Integrations and Plug-ins section.

3. On each passive node, unconfigure NNMi from the HA cluster:
   - *Windows*:

     ```
     %NnmInstallDir%\misc\nnm\ha\nnmhaunconfigure.ovpl NNM
     <resource_group>
     ```

   - *UNIX*:

     ```
     $NnmInstallDir/misc/nnm/ha/nnmhaunconfigure.ovpl NNM
     <resource_group>
     ```

   This command removes access to the shared disk but does not unconfigure the disk group or the volume group.

4 On each passive node, remove the resource group-specific files:

- *Windows*:

  In Windows Explorer, delete all files in the `%NnmDataDir%\hacluster\<resource_group>\` folder.

- *HP-UX*:

```
rm -rf $NnmDataDir/hacluster/<resource_group>/*
rm -rf /etc/cmcluster/<resource_group>/*
```

- *Linux*:

```
rm -rf $NnmDataDir/hacluster/<resource_group>/*
rm -rf /usr/local/cmcluster/conf/<resource_group>/*
```

- *Solaris*:

```
rm -rf $NnmDataDir/hacluster/<resource_group>/*
```

If you want to keep NNMi configured for high availability, stop here.

5 On the active node, unconfigure the add-on iSPIs from the HA cluster. For each iSPI, run the following command:

- *Windows*:

```
%NnmInstallDir%\misc\nnm\ha\nnmhaunconfigure.ovpl NNM
-addon <iSPI_PM_Name>
```

- *UNIX*:

```
$NnmInstallDir/misc/nnm/ha/nnmhaunconfigure.ovpl NNM
-addon <iSPI_PM_Name>
```

6 On the active node, stop the NNMi HA resource group:

- *Windows*:

```
%NnmInstallDir%\misc\nnm\ha\nnmhastoprg.ovpl NNM
<resource_group>
```

- *UNIX*:

```
$NnmInstallDir/misc/nnm/ha/nnmhastoprg.ovpl NNM
<resource_group>
```

This command does not remove access to the shared disk. Nor does it unconfigure the disk group or the volume group.

7 On the active node, remove the resource group-specific files:

- *Windows*:

  In Windows Explorer, delete all files in the `%NnmDataDir%\hacluster\<resource_group>\` folder.

- *HP-UX*:

```
rm -rf $NnmDataDir/hacluster/<resource_group>/*
rm -rf /etc/cmcluster/<resource_group>/*
```

- *Linux*:

  ```
  rm -rf $NnmDataDir/hacluster/<resource_group>/*
  rm -rf /usr/local/cmcluster/conf/<resource_group>/*
  ```

- *Solaris*:

  ```
  rm -rf $NnmDataDir/hacluster/<resource_group>/*
  ```

8 Unmount the shared disk.

- If you want to reconfigure the NNMi HA cluster at some point, you can keep the disk in its current state.
- If you want to use the shared disk for another purpose, copy all data that you want to keep (as described in the next procedure), and then use the HA product commands to unconfigure the disk group and volume group.

If you want to run NNMi outside of HA on any node with the existing database, follow these steps:

1 On the active node (if one still exists), verify that NNMi is not running:

```
ovstop
```

Alternatively, check the status of the `ovspmd` process by using Task Manager (Windows) or the `ps` command (UNIX).

2 On the current node (where you want to run NNMi outside of HA), verify that NNMi is not running:

```
ovstop
```

To prevent data corruption, ensure that no instance of NNMi or the NNM iSPI for Performance is running and accessing the shared disk.

3 Use the appropriate operating system commands to mount the shared disk. For example:

- *Windows*: Use Windows Explorer.
- *UNIX*:

  ```
  vgchange -a e vgnnm
  mount /dev/vgnnm/lvnnm /nnmmount
  ```

4 Copy the NNMi files from the shared disk to the node:

- *Windows*:

  ```
  %NnmInstallDir%\misc\nnm\ha\nnmhadisk.ovpl NNM
  -from <HA_mount_point>
  ```

- *UNIX*:

  ```
  $NnmInstallDir/misc/nnm/ha/nnmhadisk.ovpl NNM
  -from <HA_mount_point>
  ```

5 Use the appropriate operating system commands to unmount the shared disk. For example:

- *Windows*: Use Windows Explorer and Disk Management.
- *UNIX*:

  ```
  umount /nnmmount
  vgchange -a n vgnnm
  ```

6 Start NNMi:

```
ovstart -c
```

NNMi is now running with a copy of the database that was formerly used by the NNMi HA resource group. Manually remove from the NNMi configuration any nodes that you do not want to manage from this NNMi management server.

## Unconfiguring the NNM iSPI for Performance from an HA Cluster

The process of removing an NNM iSPI for Performance node from an HA cluster involves undoing the HA configuration for that instance of the iSPI. You can then run that instance of the iSPI as a standalone server, or you can uninstall the iSPI from that node.

If you want to keep the NNM iSPI for Performance configured for high availability, the HA cluster must contain one node that is actively running the iSPI and at least one passive iSPI node. If you want to completely remove the NNM iSPI for Performance from the HA cluster, unconfigure the HA functionality on all nodes in the cluster.

To completely unconfigure the NNM iSPI for Performance from an HA cluster, follow these steps:

1 Determine which node in the HA cluster is active:

- *Windows*:

```
%NnmInstallDir%\misc\nnm\ha\nnmhaclusterinfo.ovpl
-group <resource_group> -state
```

- *UNIX*:

```
$NnmInstallDir/misc/nnm/ha/nnmhaclusterinfo.ovpl
-group <resource_group> -state
```

2 On each passive node, unconfigure the NNM iSPI for Performance from the HA cluster:

- *Windows*:

```
%NnmInstallDir%\misc\nnm\ha\nnmhaunconfigure.ovpl PerfSPIHA
<resource_group>
```

- *UNIX*:

```
$NnmInstallDir/misc/nnm/ha/nnmhaunconfigure.ovpl PerfSPIHA
<resource_group>
```

This command removes access to the shared disk but does not unconfigure the disk group or the volume group.

3 On each passive node, remove the resource group-specific files:

- *Windows*:

In Windows Explorer, delete all files in the `%NnmDataDir%\hacluster\<resource_group>\` folder.

- *HP-UX*:

```
rm -rf $NnmDataDir/hacluster/<resource_group>/*
rm -rf /etc/cmcluster/<resource_group>/*
```

- *Linux*:

```
rm -rf $NnmDataDir/hacluster/<resource_group>/*
rm -rf /usr/local/cmcluster/conf/<resource_group>/*
```

- *Solaris*:

```
rm -rf $NnmDataDir/hacluster/<resource_group>/*
```

If you want to keep the NNM iSPI for Performance configured for high availability, stop here.

4 On the active node, stop the performance HA resource group:

- *Windows*:

```
%NnmInstallDir%\misc\nnm\ha\nnmhastoprg.ovpl PerfSPIHA
<resource_group>
```

- *UNIX*:

```
$NnmInstallDir/misc/nnm/ha/nnmhastoprg.ovpl PerfSPIHA
<resource_group>
```

This command does not remove access to the shared disk. Nor does it unconfigure the disk group or the volume group.

5 On the active node, remove the resource group-specific files:

- *Windows*:

  In Windows Explorer, delete all files in the `%NnmDataDir%\hacluster\<resource_group>\` folder.

- *HP-UX*:

```
rm -rf $NnmDataDir/hacluster/<resource_group>/*
rm -rf /etc/cmcluster/<resource_group>/*
```

- *Linux*:

```
rm -rf $NnmDataDir/hacluster/<resource_group>/*
rm -rf /usr/local/cmcluster/conf/<resource_group>/*
```

- *Solaris*:

```
rm -rf $NnmDataDir/hacluster/<resource_group>/*
```

6 Unmount the shared disk.

- If you want to reconfigure the performance HA cluster at some point, you can keep the disk in its current state.
- If you want to use the shared disk for another purpose, copy all data that you want to keep (as described in the next procedure), and then use the HA product commands to unconfigure the disk group and volume group.

If you want to run the NNM iSPI for Performance outside of HA on any node with the existing database, follow these steps:

1 On the active node (if one still exists), verify that the NNM iSPI for Performance is not running:

- *Windows*:

```
%NnmInstallDir%\NNMPerformanceSPI\bin\stopSPI.ovpl
%NnmInstallDir%\NNMPerformanceSPI\bin\stopERS.ovpl
```

- *UNIX*:

  **$NnmInstallDir/NNMPerformanceSPI/bin/stopSPI.ovpl**
  **$NnmInstallDir/NNMPerformanceSPI/bin/stopERS.ovpl**

2 On the current node (where you want to run the NNM iSPI for Performance outside of HA), verify that the iSPI is not running:

- *Windows*:

  **%NnmInstallDir%\NNMPerformanceSPI\bin\stopSPI.ovpl**
  **%NnmInstallDir%\NNMPerformanceSPI\bin\stopERS.ovpl**

- *UNIX*:

  **$NnmInstallDir/NNMPerformanceSPI/bin/stopSPI.ovpl**
  **$NnmInstallDir/NNMPerformanceSPI/bin/stopERS.ovpl**

To prevent data corruption, ensure that no instance of the iSPI is running and accessing the shared disk.

3 Use the appropriate operating system commands to mount the shared disk. For example:

- *Windows*: Use Windows Explorer.
- *UNIX*:

```
vgchange -a e vgnnm
mount /dev/vgnnm/lvnnm /nnmmount
```

4 Copy the iSPI files from the shared disk to the node:

- *Windows*:

  **%NnmInstallDir%\misc\nnm\ha\nnmhadisk.ovpl PerfSPIHA -from <*HA_mount_point*>**

- *UNIX*:

  **$NnmInstallDir/misc/nnm/ha/nnmhadisk.ovpl PerfSPIHA -from <*HA_mount_point*>**

5 Use the appropriate operating system commands to unmount the shared disk. For example:

- *Windows*: Use Windows Explorer and Disk Management.
- *UNIX*:

```
umount /nnmmount
vgchange -a n vgnnm
```

6 Start the NNM iSPI for Performance:

- *Windows*:

  **%NnmInstallDir%\NNMPerformanceSPI\bin\startSPI.ovpl**
  **%NnmInstallDir%\NNMPerformanceSPI\bin\startERS.ovpl**

- *UNIX*:

  **$NnmInstallDir/NNMPerformanceSPI/bin/startSPI.ovpl**
  **$NnmInstallDir/NNMPerformanceSPI/bin/startERS.ovpl**

The NNM iSPI for Performance is now running with a copy of the database that was formerly used by the performance HA resource group.

# Upgrading NNMi under HA from NNMi 8.0x to NNMi 8.10

To upgrade from NNMi 8.0x under HA to NNMi 8.10 under HA, unconfigure NNMi 8.0x from HA, install NNMi 8.10 to get the updated database schema, and then reconfigure NNMi for HA. Follow these steps:

1. Verify that the NNMi 8.0x configuration is consistent across all HA nodes:
   - *Windows*:

     ```
     %NnmInstallDir%\misc\nnm\ha\nnmdatareplicator.ovpl NNM
     ```

   - *UNIX*:

     ```
     $NnmInstallDir/misc/nnm/ha/nnmdatareplicator.ovpl NNM
     ```

2. Determine which node in the NNMi 8.0x HA cluster is active:
   - *Windows*:

     ```
     %NnmInstallDir%\misc\nnm\ha\nnmhaclusterinfo.ovpl
     -group <resource_group> -state
     ```

   - *UNIX*:

     ```
     $NnmInstallDir/misc/nnm/ha/nnmhaclusterinfo.ovpl
     -group <resource_group> -state
     ```

3. On each passive node, unconfigure NNMi 8.0x:
   a. Unconfigure NNMi from the HA cluster:
      — *Windows*:

      ```
      %NnmInstallDir%\misc\nnm\ha\nnmhaunconfigure.ovpl
      <resource_group>
      ```

      — *UNIX*:

      ```
      $NnmInstallDir/misc/nnm/ha/nnmhaunconfigure.ovpl
      <resource_group>
      ```

      This command removes access to the shared disk but does not unconfigure the disk group or the volume group.

   b. Remove the resource group-specific files:
      — *Windows*:

      In Windows Explorer, delete all files in the `%NnmDataDir%\hacluster\<resource_group>\` folder.

      — *HP-UX*:

      ```
      rm -rf $NnmDataDir/hacluster/<resource_group>/*
      rm -rf /etc/cmcluster/<resource_group>/*
      ```

      — *Linux*:

      ```
      rm -rf $NnmDataDir/hacluster/<resource_group>/*
      rm -rf /usr/local/cmcluster/conf/<resource_group>/*
      ```

      — *Solaris*:

      ```
      rm -rf $NnmDataDir/hacluster/<resource_group>/*
      ```

4 On the active node, unconfigure NNMi 8.0x:

a Disable HA resource group monitoring by creating the following maintenance file:

— *Windows*: `%NnmDataDir%\hacluster\maint_NNM`

— *UNIX*: `$NnmDataDir/hacluster/<resource_group>/maint_NNM`

The file can be empty.

b Stop NNMi:

```
ovstop -c
```

To prevent data corruption, ensure that no instance of NNMi is running and accessing the shared disk.

c Back up the NNMi database that is on the shared disk:

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

For more information on this command, see NNMi Backup and Restore Tools on page 123.

d Copy the NNMi files from the shared disk to the node:

— *Windows*:

```
%NnmInstallDir%\misc\nnm\ha\nnmhadisk.ovpl
-from <HA_mount_point>
```

— *UNIX*:

```
$NnmInstallDir/misc/nnm/ha/nnmhadisk.ovpl
-from <HA_mount_point>
```

e Remove all NNMi files and directories from the shared disk:

— *Windows*: Use Windows Explorer to delete all files under the shared disk mount point (`%HA_MOUNT_POINT%`, which is, for example `S:\`).

— *UNIX*:

```
rm -rf $HA_MOUNT_POINT/*
```

f Stop the NNMi HA resource group:

— *Windows*:

```
%NnmInstallDir%\misc\nnm\ha\nnmhastoprg.ovpl NNM
<resource_group>
```

— *UNIX*:

```
$NnmInstallDir/misc/nnm/ha/nnmhastoprg.ovpl NNM
<resource_group>
```

g Unconfigure NNMi from the HA cluster:

— *Windows*:

```
%NnmInstallDir%\misc\nnm\ha\nnmhaunconfigure.ovpl
<resource_group>
```

— *UNIX*:

```
$NnmInstallDir/misc/nnm/ha/nnmhaunconfigure.ovpl
<resource_group>
```

This command removes access to the shared disk but does not unconfigure the disk group or the volume group.

h Remove the resource group-specific files:

— *Windows*:

In Windows Explorer, delete all files in the `%NnmDataDir%\hacluster\<resource_group>\` folder.

— *HP-UX*:

```
rm -rf $NnmDataDir/hacluster/<resource_group>/*
rm -rf /etc/cmcluster/<resource_group>/*
```

— *Linux*:

```
rm -rf $NnmDataDir/hacluster/<resource_group>/*
rm -rf /usr/local/cmcluster/conf/<resource_group>/*
```

— *Solaris*:

```
rm -rf $NnmDataDir/hacluster/<resource_group>/*
```

i Delete the maintenance file:

— *Windows*: `%NnmDataDir%\hacluster\maint_NNM`

— *UNIX*: `$NnmDataDir/hacluster/<resource_group>/maint_NNM`

5 On the node that was active before unconfiguring NNMi 8.0x from HA, configure NNMi 8.10:

a Install the NNMi 8.10 upgrade.

The upgrade process starts NNMi to update the NNMi database schema.

b Follow the steps in Configuring NNMi on the Primary Cluster Node on page 86.

You do not need to do the following:

— Define a disk device group and logical volume.

— Create a mount point for the shared disk.

— Configure the shared disk.

6 On each node that was passive before unconfiguring NNMi 8.0x from HA, configure NNMi 8.10:

a Install the NNMi 8.10 upgrade.

b On the first secondary node, follow the steps in Configuring NNMi on the Secondary Cluster Nodes on page 88.

You do not need to create a mount point for the shared disk.

# Troubleshooting the HA Configuration

## General HA Troubleshooting

The topics in this section apply to HA configuration for all HP NNM i-series Software products.

### Error: Wrong Number of Arguments

As of NNMi 8.10, the name of the product Perl module is a required parameter to most of the NNMi HA configuration commands.

- For NNMi use the value `NNM`.
- To determine what value to use for an iSPI, see Table 6 on page 89 or the appropriate iSPI chapter in the Integrations and Plug-ins section.

### Product Startup Times Out (Solaris)

One or more of the `/var/adm/messages*` files contains a message similar to the following example:

```
VCS ERROR V-16-1-13012 Thread(…) Resource(<resource group>-app):
online procedure did not complete within the expected time.
```

This message indicates that the product did not start completely within the Veritas timeout value. The NNMi-provided HA configuration scripts define this timeout to be 15 minutes.

To change the Veritas timeout value on the Solaris operating system, run the following commands in order:

```
/opt/VRTSvcs/bin/haconf -makerw
/opt/VRTSvcs/bin/hares -modify <resource_group>-app OnlineTimeout <value in seconds>
/opt/VRTSvcs/bin/haconf -dump -makero
```

### Log Files on the Active Cluster Node Are Not Updating

This situation is normal. It occurs because the log files have been redirected to the shared disk.

For NNMi, review the log files in the location specified by `HA_NNM_LOG_DIR` in the `ov.conf` file.

For the NNM iSPI for Performance, the local log files are removed during HA configuration. For the location of the iSPI log files on the shared disk, see NNM iSPI for Performance HA Log Files on page 121.

## Cannot Start the HA Resource Group on a Particular Cluster Node

If the `nnmhargconfigure.ovpl` command does not correctly start, stop, or switch the NNMi HA resource group, review the following information:

- *Windows*:
  - In Cluster Administrator, review the state of the resource group and underlying resources.
  - Review the Event Viewer log for any errors.
- *HP-UX*:

  Review the `/etc/cmcluster/<resource_group>/<resource_group>.cntl.log` file and the syslog files for errors. The most common problems are leaving the system in a state where a resource cannot be added, for example, having a disk group misconfigured such that it cannot be activated.
- *Linux*:

  Review the `/usr/local/cmcluster/conf/<resource_group>/<resource_group>.cntl.log` file and the syslog files for errors. The most common problems are leaving the system in a state where a resource cannot be added, for example, having a disk group misconfigured such that it cannot be activated.
- *Solaris*:
  - Run **`/opt/VRTSvcs/bin/hares -state`** to review the resource state.
  - For failed resources, review the `/var/VRTSvcs/log/<resource>.log` file for the resource that is failing. Resources are referenced by the agent type, for example: `IP*.log`, `Mount*.log`, and `Volume*.log`.

If you cannot locate the source of the problem, you can manually start the HA resource group by using the HA product commands:

1. Mount the shared disk.
2. Assign the virtual host to the network interface:
   - *Windows*:
     - Start Cluster Administrator.
     - Expand the resource group.
     - Right-click ***<resource_group>*-ip**, and then click **Bring Online**.
   - *HP-UX*: Run `/usr/sbin/cmmodnet` to add the IP address.
   - *Linux*: Run `/usr/local/cmcluster/bin/cmmodnet` to add the IP address.
   - *Solaris*: **`/opt/VRTSvcs/bin/hares -online <resource_group>-ip -sys <local_hostname>`**

3 Start the HA resource group. For example:

- *Windows*:

```
%NnmInstallDir%\misc\nnm\ha\nnmhastart.ovpl NNM
-start <resource_group>
```

- *UNIX*:

```
$NnmInstallDir/misc/nnm/ha/nnmhastart.ovpl NNM
-start <resource_group>
```

The return code `0` indicates that NNMi started successfully.

The return code `1` indicates that NNMi did not start correctly.

## NNMi-Specific HA Troubleshooting

The topics in this section apply to HA configuration for NNMi only.

### Re-Enable NNMi for HA after All Cluster Nodes are Unconfigured

When all NNMi HA cluster nodes have been unconfigured, the `ov.conf` file no longer contains any mount point references to the NNMi shared disk. To re-create the mount point reference without overwriting the data on the shared disk, follow these steps on the primary node:

1 If NNMi is running, stop it:

```
ovstop -c
```

2 Reset the reference to the shared disk:

- *Windows*:

```
%NnmInstallDir%\misc\nnm\ha\nnmhadisk.ovpl NNM
-setmount <HA_mount_point>
```

- *UNIX*:

```
$NnmInstallDir/misc/nnm/ha/nnmhadisk.ovpl NNM
-setmount <HA_mount_point>
```

3 In the `ov.conf` file, verify the entries related to HA mount points.

For the location of the `ov.conf` file, see NNMi HA Configuration Files on page 118.

### NNMi Does Not Start Correctly Under HA

Either of the following situations occurs:

- `ovspmd` does not start after NNMi is configured for HA. (`ovstatus` shows that `ovspmd` is NOT RUNNING.)
- `ovstart` or `ovstop` returns the following message:

```
ovstart: Must specify managers by name if running on OVW
Client
```

While NNMi is running under HA, *do not* use the `ovstart` and `ovstop` commands unless instructed to do so for HA maintenance purposes.

The NNMi configuration points to a different system than where NNMi is running. To fix the problem, verify that the `ov.conf` file has appropriate entries for the following items:

- `NNM_INTERFACE=<virtual_hostname>`
- `NNM_HA_CONFIGURED=YES`
- `HA_RESOURCE_GROUP=<resource_group>`
- `HA_POSTGRES_DIR=<HA_mount_point>/NNM/dataDir/shared/nnm/databases/Postgres`
- `HA_EVENTDB_DIR=<HA_mount_point>/NNM/dataDir/shared/nnm/eventdb`
- `HA_NNM_LOG_DIR=<HA_mount_point>/NNM/dataDir/log`
- `HA_JBOSS_DATA_DIR=<HA_mount_point>/NNM/installDir/nonOV/jboss/nms/server/nms/data`
- `HA_MOUNT_POINT=<HA_mount_point>`

For the location of the `ov.conf` file, see NNMi HA Configuration Files on page 118.

## nmsdbmgr Does Not Start after HA Configuration

This situation usually occurs as a result of starting NNMi immediately after running the `nnmhadisk.ovpl` command with the `-to` option. (The `nnmhaconfigure.ovpl` command has not been run.) In this case, the `HA_POSTGRES_DIR` entry in the `ov.conf` file specifies the location of the embedded database on the shared disk, but this location is not available to NNMi.

Run `nnmhaconfigure.ovpl`, and then complete the HA configuration.

## pmd Does Not Start after HA Configuration

This situation usually occurs after a configuration error such as not setting up the shared disk correctly. The failure of the `pmd` process occurs when the `ovjboss` process does not fully start.

Review the following log file:

- *Windows*: `%HA_MOUNT_POINT%\NNM\dataDir\log\nnm\jbossServer.log`
- *UNIX*: `$HA_MOUNT_POINT/NNM/dataDir/log/nnm/jbossServer.log`

## Disk Failover Does Not Occur

This situation can happen when the shared disk is not supported for the appropriate platform. Review the appropriate HA product, operating system, and disk manufacturer documentation to determine whether these products can all work together.

If disk failure occurs, NNMi will not start upon failover. Most likely, `nmsdbmgr` will fail because the `HA_POSTGRES_DIR` directory does not exist. Verify that the shared disk is mounted and that the appropriate files are accessible.

### Shared Disk Files Are Not Found on the Secondary Node after Failover

The most common cause of this situation is that the `nnmhadisk.ovpl` command was run with the `-to` option when the shared disk was not mounted. In this case, the data files are copied to the local disk, so the files are not available on the shared disk.

## iSPI-Specific HA Troubleshooting

The topics in this section apply to add-on HA configuration for all HP NNM i-series Software iSPIs.

### iSPIs Do Not Start Correctly Under HA

Call HP Support for assistance before you begin this procedure.

1. On the cluster node in question, determine whether each iSPI is correctly registered by examining the current values for the `NNM_ADD_ON_PRODUCTS` attribute:
   - *Windows*: Run the following command:

     ```
     cluster group <resource_group> /PRIV
     ```

     In the output, look for `PUBLIC.NNM_ADD_ON_PRODUCTS`.
   - *HP-UX*: Look in the `/etc/cmcluster/<resource_group>/<resource_group>.public.env` file.
   - *Linux*: Look in the `/usr/local/cmcluster/conf/<resource_group>/<resource_group>.public.env` file.
   - *Solaris*: Stored as part of the resource group specification in `UserStrGlobal`.
2. If the values for `NNM_ADD_ON_PRODUCTS` are not correct, update the values:
   - *Windows*: Run the following command:

     ```
     cluster group <resource_group> /PRIV
     <attribute_name>=<value>
     ```

Do not edit the Windows Registry directly.

   - *HP-UX*: Copy a good version of the `/etc/cmcluster/<resource_group>/<resource_group>.public.env` file from another cluster node to this one.
   - *Linux*: Copy a good version of the `/usr/local/cmcluster/conf/<resource_group>/<resource_group>.public.env` file from another cluster node to this one.
   - *Solaris*: Update `UserStrGlobal`:

     ```
     /opt/VRTSvcs/bin/haconf -makerw
     /opt/VRTSvcs/bin/hagrp -modify <resource_group>
     UserStrGlobal "<newvalue>"
     ```

# HA Configuration Reference

## NNMi HA Configuration Files

Table 8 lists the NNMi HA configuration files. These files apply to NNMi and add-on iSPIs on the NNMi management server. These files are installed to the following location:

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

**Table 8 NNMi HA Configuration Files**

| File Name | Description |
|---|---|
| ov.conf | Updated by the `nnmhaclusterinfo.ovpl` command to describe the NNMi HA implementation. NNMi processes read this file to determine the HA configuration. |
| nnmdatareplicator.conf | Used by the `nnmdatareplicator.ovpl` command to determine which NNMi folders and files are included in data replication from the active node to the passive nodes. If you implement a different method of replicating the NNMi configuration, see this file for a list of the data to include.<br>For more information, see the comments in the file. |

## NNM iSPI for Performance HA Configuration Files

The add-on NNM iSPI for Performance uses the `ov.conf` file for configuring HA.

The NNM iSPI for Performance on a dedicated server uses the following file for configuring HA:

- *Windows*:
  - `%NnmDataDir%\shared\perfSpi\conf\perfspi.conf`
- *UNIX*:
  - `$NnmDataDir/shared/perfSpi/conf/perfspi.conf`

## NNMi-Provided HA Configuration Scripts

Table 9 lists the HA configuration scripts that are included with NNMi. The NNMi-provided scripts are convenience scripts that can be used to configure HA for any product that has a customer Perl module. If you prefer, you can use the HA product-provided commands to configure HA for NNMi.

On the NNMi management server, the NNMi-provided HA configuration scripts are installed to the following location:

- *Windows*: `%NnmInstallDir%\misc\nnm\ha`
- *UNIX*: `$NnmInstallDir/misc/nnm/ha`

On the HP-UX and Solaris operating systems, this directory contains obsoleted scripts (with names `ov*`). Do not run these scripts.

**Table 9 NNMi HA Configuration Scripts**

| Script Name | Description |
|---|---|
| nnmhaconfigure.ovpl | Configures NNMi or an iSPI for an HA cluster.<br>Run this script on all nodes in the HA cluster. |
| nnmhaunconfigure.ovpl | Unconfigures NNMi or an iSPI from an HA cluster.<br>Optionally, run this script on one or more nodes in the HA cluster. |
| nnmhaclusterinfo.ovpl | Retrieves cluster information regarding NNMi.<br>Run this script as needed on any node in the HA cluster. |
| nnmhadisk.ovpl | Copies NNMi and iSPI data files to and from the shared disk.<br>During HA configuration, run this script on the primary node.<br>At other times, run this script per the instructions in this chapter. |
| nnmhastartrg.ovpl | Starts NNMi in an HA cluster.<br>During HA configuration, run this script on the primary node. |
| nnmdatareplicator.ovpl | Checks the `nnmdatareplicator.conf` configuration file for changes and copies files to remote systems. |
| nnmharg.ovpl | Starts, stops, and monitors NNMi in an HA cluster.<br>Used by `<resource_group>.cntl` for ServiceGuard configurations (HP-UX and Linux).<br>Used by the Veritas start, stop, and monitor scripts on Solaris. (`nnmhargconfigure.ovpl` configures this usage.)<br>Also used by `nnmhastartrg.ovpl` to enable and disable tracing. |
| nnmhargconfigure.ovpl | Configures HA resources and resource groups. Used by `nnmhaconfigure.ovpl` and `nnmhaunconfigure.ovpl`. |
| nnmhastart.ovpl | Starts NNMi in an HA cluster. Used by `nnmharg.ovpl`. |
| nnmhastop.ovpl | Stops NNMi in an HA cluster. Used by `nnmharg.ovpl`. |
| nnmhamonitor.ovpl | Monitors NNMi processes in an HA cluster. Used by `nnmharg.ovpl`. |
| nnmhamscs.vbs | Is a template for creating a script to start, stop, and monitor NNMi processes in a Microsoft Cluster Services HA cluster. The generated script is used by Microsoft Cluster Services and is stored in the following location:<br>`%NnmDataDir%\hacluster\<resource_group>\hamscs.vbs` |

## NNMi HA Configuration Log Files

The following log files apply to the HA configuration for NNMi and add-on iSPIs on the NNMi management server:

- *Windows* configuration:
  - `%NnmDataDir%\tmp\HA_nnmhaserver.log`
  - `%NnmDataDir%\log\haconfigure.log`
- *UNIX* configuration:
  - `$NnmDataDir/tmp/HA_nnmhaserver.log`
  - `$NnmDataDir/log/haconfigure.log`
- *Windows* runtime:
  - Event Viewer log
  - `%HA_MOUNT_POINT%\NNM\dataDir\log\nnm\ovspmd.log`
  - `%HA_MOUNT_POINT%\NNM\dataDir\log\nnm\public\postgres.log`
  - `%HA_MOUNT_POINT%\NNM\dataDir\log\nnm\public\nmsdbmgr.log`
  - `%HA_MOUNT_POINT%\NNM\dataDir\log\nnm\jbossServer.log`
  - `%HA_MOUNT_POINT%\NNM\dataDir\log\nnm\ovet*.log`
  - `%SystemRoot%\Cluster\cluster.log`
    This is the log file for cluster runtime issues including: adding and removing resources and resource groups; other configuration issues; starting and stopping issues.
- *HP-UX* runtime:
  - `/etc/cmcluster/<resource_group>/<resource_group>.cntl.log`
    This is the log file for the resource group.
  - `/var/adm/syslog/syslog.log`
  - `/var/adm/syslog/OLDsyslog.log`
  - `$HA_MOUNT_POINT/NNM/dataDir/log/nnm/ovspmd.log`
  - `$HA_MOUNT_POINT/NNM/dataDir/log/nnm/public/postgres.log`
  - `$HA_MOUNT_POINT/NNM/dataDir/log/nnm/public/nmsdbmgr.log`
  - `$HA_MOUNT_POINT/NNM/dataDir/log/nnm/jbossServer.log`
  - `$HA_MOUNT_POINT/NNM/dataDir/log/nnm/ovet*.log`
- *Linux* runtime:
  - `/usr/local/cmcluster/conf/<resource_group>/<resource_group>.cntl.log`
    This is the log file for the resource group.
  - `/var/log/cmcluster`
    This is the log file for cluster issues.
  - `/var/log/messages*`
    These are the log files for resource issues (disk and IP address).
  - `$HA_MOUNT_POINT/NNM/dataDir/log/nnm/ovspmd.log`
  - `$HA_MOUNT_POINT/NNM/dataDir/log/nnm/public/postgres.log`

— `$HA_MOUNT_POINT/NNM/dataDir/log/nnm/public/nmsdbmgr.log`

— `$HA_MOUNT_POINT/NNM/dataDir/log/nnm/jbossServer.log`

— `$HA_MOUNT_POINT/NNM/dataDir/log/nnm/ovet*.log`

- *Solaris* runtime:

| Resource | Log File |
|---|---|
| <*resource_group*>-app | • `/var/VRTSvcs/log/Application_A.log`<br>• `$HA_MOUNT_POINT/NNM/dataDir/log/nnm/ovspmd.log`<br>• `$HA_MOUNT_POINT/NNM/dataDir/log/nnm/public/postgres.log`<br>• `$HA_MOUNT_POINT/NNM/dataDir/log/nnm/public/nmsdbmgr.log`<br>• `$HA_MOUNT_POINT/NNM/dataDir/log/nnm/jbossServer.log`<br>• `$HA_MOUNT_POINT/NNM/dataDir/log/nnm/ovet*.log`<br>• `/var/adm/messages*` |
| <*resource_group*>-dg<br><*resource_group*>-volume<br><*resource_group*>-mount | • `/var/VRTSvcs/log/DiskGroup_A.log`<br>• `/var/VRTSvcs/log/Volume_A.log`<br>• `/var/VRTSvcs/log/Mount_A.log`<br>• `/var/adm/messages*` |
| <*resource_group*>-ip | • `/var/VRTSvcs/log/IP_A.log`<br>• `/var/adm/messages*` |
| For operating system-specific issues related to the HA resources, review the `/var/adm/messages*` files. For <*resource_group*>-app, look for messages regarding unable to start process. | |

## NNM iSPI for Performance HA Log Files

The performance HA resource group uses many of the log files listed in NNMi HA Configuration Log Files on page 120. Additionally, the configuration maintains the following log files:

- *Windows* configuration:

  — `%NnmDataDir%\NNMPerformanceSPI\logs\prspiHA.log`

- *UNIX* configuration:

  — `$NnmDataDir/NNMPerformanceSPI/logs/prspiHA.log`

- *Windows* runtime:

  — `%HA_MOUNT_POINT%\NNMPerformanceSPI\dataDir\NNMPerformanceSPI\logs\*.log`

- *UNIX* runtime:

  — `$HA_MOUNT_POINT/NNMPerformanceSPI/dataDir/NNMPerformanceSPI/logs/*.log`

# NNMi Backup and Restore Tools

This chapter contains the following topics:


- About NNMi Data Backup
- Limitations of the NNMi Backup and Restore Scripts
- Backing up NNMi Data
- Restoring NNMi Data
- Backing up and Restoring the Embedded Database Only


## About NNMi Data Backup

A good backup and restore strategy is key to ensuring the uninterrupted operations of any business. HP Network Node Manager i Software is an important asset for network operations and should be backed up regularly.

HP recommends that you perform a complete weekly backup of all NNMi data. You do not need to shut down NNMi in order to create this backup. Perform scoped backups (as described in Backing up NNMi Data) as needed before beginning configuration changes. In this way, if your configuration changes do not have the expected effect, you will be able to revert to a known working configuration.

The two types of critical data related to an NNMi installation are as follows:

- Files in the file system
- Data in the relational database (embedded or external)

This chapter explains the tools that NNMi provides for backing up and restoring important NNMi files and data. If you use the NNMi embedded database, these tools might be sufficient for backing up and restoring the NNMi installation. If you use an external database (such as Oracle), read this document to identify the NNMi files and database tables that should be added to your existing database backup procedures.

**Script names**

NNMi provides the following scripts for backing up and restoring NNMi data:

- `nnmbackup.ovpl`—Backs up all necessary file system data (including configuration information) and any data stored in the embedded database.
- `nnmrestore.ovpl`—Restores a backup that was created by using the `nnmbackup.ovpl` script.
- `nnmbackupembdb.ovpl`—Creates a complete backup of the NNMi embedded database (but not the file system data) while NNMi is running.
- `nnmrestoreembdb.ovpl`—Restores a backup that was created by using the `nnmbackupembdb.ovpl` script.

**Interaction with iSPIs**

Data from Smart Plug-ins (iSPIs) can be included in the NNMi backups and restores. For information about a particular iSPI, see the appropriate chapter in the Integrations and Plug-ins section of this guide.

# Limitations of the NNMi Backup and Restore Scripts

## Backup and Restore Systems Must Be of the Same Configuration

You can use the backup and restore scripts to transfer data from one NNMi management server to another. The following items must be identical on both systems:

- NNMi version (including any patches)
- Operating system type and version
- Character set (language)

The following items can differ between the two systems:

- Hostname
- IP address

## External Databases Are Not Supported

The NNMi backup and restore scripts do not work with data stored in an external database (such as Oracle). External database maintenance should be handled as part of the existing database backup and restore procedures.

# Backing up NNMi Data

The NNMi-provided backup script copies the synchronized NNMi data to the specified target directory. You can then use any appropriate tool to save a copy of the backup.

The NNMi backup script supports two types of backups:

- Online backups occur while NNMi is running. At a minimum, the nmsdbmgr service must be running.
- Offline backups occur while NNMi is completely stopped.

You can use the NNMi backup script to create a complete backup of the NNMi data or to back up only some of the data according to function. A partial backup is useful for moving information between systems and for archiving the current state of NNMi.

Partial backups are only available for online backups. If you run the backup script while NNMi is completely stopped, the script copies all data in the embedded database regardless of the backup scope.

The syntax for the backup script command is as follows. Table 10 describes the script command line options.

```
nnmbackup.ovpl [-type (online|offline)]
   [-scope (config|topology|events|all)] [-force] [-archive]
   -target <directory>]
```

**Table 10 nnmbackup.ovpl Command Line Options**

| Option | Description |
|---|---|
| -type (online \| offline) | Specifies the type of backup. For an offline backup, stop NNMi prior to running the command or specify the `-force` option.<br>If no value is specified, an online backup is performed. |
| -scope (config \| topology \| events \| all) | Specifies the data to back up.<br>• For file system data, the scope applies to both online and offline backups.<br>• For database tables, the scope applies to online backups only. For offline backups, the entire embedded database is backed up regardless of the scope.<br>If no value is specified, a complete backup is performed. |

**Table 10 nnmbackup.ovpl Command Line Options**

| Option | Description |
|---|---|
| -force | Forces NNMi into the appropriate state for the type of backup.<br>For an online backup, the command starts the nmsdbmgr service if it is not already running.<br>For an offline backup, the command stops all NNMi services.<br>If this option is not specified and NNMi is not in the correct state for the specified backup type, the script fails. |
| -archive | Creates an archive file (tar file) in the target directory. |
| -target <*directory*> | Specifies the directory to contain the backup files. |

## Configuration scope

The configuration scope (`-scope config`) includes the files and database tables that store NNMi configuration information only. The configuration scope loosely aligns to the information in the **Configuration** workspace of the NNMi console.

Table 11 identifies the files and directories that are backed up for the configuration scope.

**Table 11 Configuration Files and Directories**

| Directory or File name | Description |
|---|---|
| $NnmInstallDir/conf (Windows only) | Configuration information |
| $NnmDataDir/conf | Configuration that might be shared by other HP products |
| $NnmDataDir/shared/nnm/conf | Shared configuration data within NNMi |
| $NnmDataDir/shared/nnm/lrf | Shared configuration data within NNMi— component registration files |
| $NnmDataDir/shared/nnm/databases/Postgres | File system storage for the NNMi embedded database<br>**NOTE**: This directory is only backed up for offline backups. |
| $NnmDataDir/NNMVersionInfo | NNMi version information file |
| $NnmInstallDir/misc/nnm | Miscellaneous configuration data |
| $NnmInstallDir/newconfig | Installation configuration staging area |
| $NnmInstallDir/nonOV/jboss/nms/server/nms/conf | jboss configuration |

**Table 11 Configuration Files and Directories (cont'd)**

| Directory or File name | Description |
|---|---|
| $NnmInstallDir/nonOV/jboss/nms/server/nms/deploy | jboss deployment directory |
| $NnmInstallDir/snmp_mibs (Windows only)<br>$NnmDataDir/share/snmp_mibs (UNIX only) | SNMP MIB information |
| $NnmDataDir/HPOvLIC/LicFile.txt | License information |

### Topology scope

The topology scope (`-scope topology`) includes the files and database tables that store NNMi network topology information. The topology scope loosely aligns to the information in the **Inventory** workspace of the NNMi console.

Because the network topology is dependent on the configuration that was used for discovering that topology, the topology scope includes the configuration scope.

### Event scope

The event scope (`-scope event`) includes the files and database tables that store NNMi network incident and event information. The event scope loosely aligns to the information in the **Incident Browsing** workspace of the NNMi console.

Because events are dependent on the network topology related to those events, the event scope includes the configuration and topology scopes.

### Complete backup

The complete backup (`-scope all`) includes all important NNMi files and data. It is the conjunction of the configuration, topology, and event scopes.

## Restoring NNMi Data

The NNMi restore script copies the data from the specified source directory to the appropriate locations in the file system and the embedded database. You can restore data from a partial backup or from a complete backup.

- The NNMi restore script only works with the information collected by the NNMi backup script. Copy the saved backup data to a directory that is accessible from the NNMi management server, and then run the restore script to copy that data to the correct locations.
- Because the database schema might change from one version of NNMi to the next, data backups cannot be shared across versions of NNMi.

Stop all NNMi services before running the restore script. If you are restoring data from an online backup, use the `-force` option

During a restoration, the target database is completely cleaned out and replaced with the contents of the backup. This cleaning can be useful for things like moving configuration information from one system to another (perhaps because of new hardware).

If the provided source is a tar file, the restore command extracts the tar file to a temporary folder in the current working directory. In this case, either ensure that the current working directory has adequate storage to support the temporary folder, or extract the archive before running the restore command.

If you are restoring a backup from a different NNMi management server, check the license information from the NNMi console login page to determine whether the license information is correct for the new system. If the restored license is not valid for the new NNMi management server, obtain and apply a new license for the new NNMi management server. For more information, see the licensing documentation.

The syntax for the restore script command is as follows. Table 12 describes the script command line options.

```
nnmrestore.ovpl [-force] -source <directory>
```

**Table 12 nnmrestore.ovpl Command Line Options**

| Option | Description |
|---|---|
| -force | Forces NNMi into the appropriate state for the type of restoration.<br>For all restorations, the command stops all NNMi services.<br>For a restoration of an online backup, the command starts the nmsdbmgr service at the appropriate point in the restoration process. |
| -source <*directory*> | Specifies the directory that contains the backup files. |

# Backing up and Restoring the Embedded Database Only

NNMi provides scripts to backup and restore the embedded database only, which is useful for creating a snapshot of the data as you experiment with NNMi configuration settings. The embedded database backup and restore scripts perform online backups only. At a minimum, the nmsdbmgr service must be running.

Best practice

Run the embedded database reset script (`nnmresetembdb.ovpl`) before restoring data to the embedded database. This script ensures that the database does not contain any errors, thereby eliminating the possibility of encountering database constraint violations. For information about running the embedded database reset script, see the `nnmresetembdb.ovpl` reference page, or the UNIX manpage.

The syntax for the embedded database backup script command is as follows. Table 13 describes the script command line options.

```
nnmbackupembdb.ovpl [-force] -target <file>
```

**Table 13 nnmbackupembdb.ovpl Command Line Options**

| Option | Description |
|---|---|
| -force | Starts the nmsdbmgr service if it is not already running.<br>If this option is not specified and the nmsdbmgr service is not running, the script fails. |
| -target <*file*> | Specifies the name of the backup file. |

The syntax for the embedded database restore script command is as follows. Table 14 describes the script command line options.

```
nnmrestoreembdb.ovpl [-force] -source <file>
```

**Table 14 nnmrestoreembdb.ovpl Command Line Options**

| Option | Description |
|---|---|
| -force | Starts the nmsdbmgr service if it is not already running.<br>If this option is not specified and the nmsdbmgr service is not running, the script fails. |
| -source <*file*> | Specifies the name of the file that contains the backup. |

# Changing the NNMi Management Server

You can duplicate the HP Network Node Manager i Software configuration on another system, for example, to move from a test environment to a production environment or to change the hardware of the NNMi management server.

You can change the IP address of the NNMi management server without affecting the NNMi configuration.

This chapter contains the following topics:

- Best Practices for Preparing the NNMi Configuration to Be Moved
- Moving the NNMi Configuration
- Changing the IP Address of an NNMi Management Server

## Best Practices for Preparing the NNMi Configuration to Be Moved

The following best practices apply to moving the NNMi configuration to a different system:

- If the node group configuration uses hostnames to identify managed nodes, the production and test NNMi management servers must use the same DNS servers. In the case that the production and test systems use different DNS servers, changes in the resolved name for a managed node might result in different polling settings between the two NNMi management servers.
- You can limit the configuration export to a single author. Create a new author value that is unique to your group or company. Specify this author value when you create or modify any of the following items:
  - Device profile
  - Incident configuration
  - URL action
- If you plan to install Smart Plug-ins (iSPIs), see the appropriate chapters in the Integrations and Plug-ins section.

# Moving the NNMi Configuration

Use the `nnmconfigexport.ovpl` command to output the NNMi configuration to an XML file. Then, use the `nnmconfigimport.ovpl` command to pull this configuration from the XML file into NNMi on the new system.

For information about these commands, refer to the appropriate reference pages, or the UNIX manpages.

You can only move the NNMi configuration. HP does not support moving topology or incident data from one NNMi management server to a different NNMi management server. Nor does HP support moving iSPI data, such as performance data that was collected for the NNM iSPI for Performance.

# Changing the IP Address of an NNMi Management Server

If you need to change the IP address of the NNMi management server, follow these steps:

1. Go to **http://www.webware.hp.com**.
2. Click **Manage Licenses**.
3. Log in, and then follow the procedures to complete the move process to obtain your new license key.
4. Configure the NNMi management server with the new IP address.
5. Configure your DNS servers to recognize the new IP address of the NNMi management server.
6. Reboot the NNMi management server.
7. At a command prompt, enter the following command:

   ```
   nnmlicense.ovpl NNM -g
   ```

8. In the **Autopass: License Management** dialog box, click **Remove License Key**.
9. Select the license key to remove.
10. Select **Remove Licenses permanently**.
11. Click **Remove**, and then close the dialog box.
12. Copy the new license key that you obtained in step 3 into a text file named `license.txt`.
13. Run the following command:

    ```
    nnmlicense.ovpl NNM -f license.txt
    ```

# Migration from NNM 6.x/7.x

This section contains the following chapters:


- Product Comparison
- Migrating from NNM 6.x/7.x
- Integrating NNM 6.x or NNM 7.x with NNMi 8.10

# Product Comparison

This chapter describes the key differences between HP Network Node Manager (NNM) 6.x/7.x and HP Network Node Manager i Software. If you are familiar with a prior version of NNM, refer to this chapter as you plan and configure NNMi. If you are new to NNMi, you do not need to read this chapter.

This chapter contains the following topics:


- Network Discovery
- Status Monitoring
- Customizing Event Monitoring

# Network Discovery

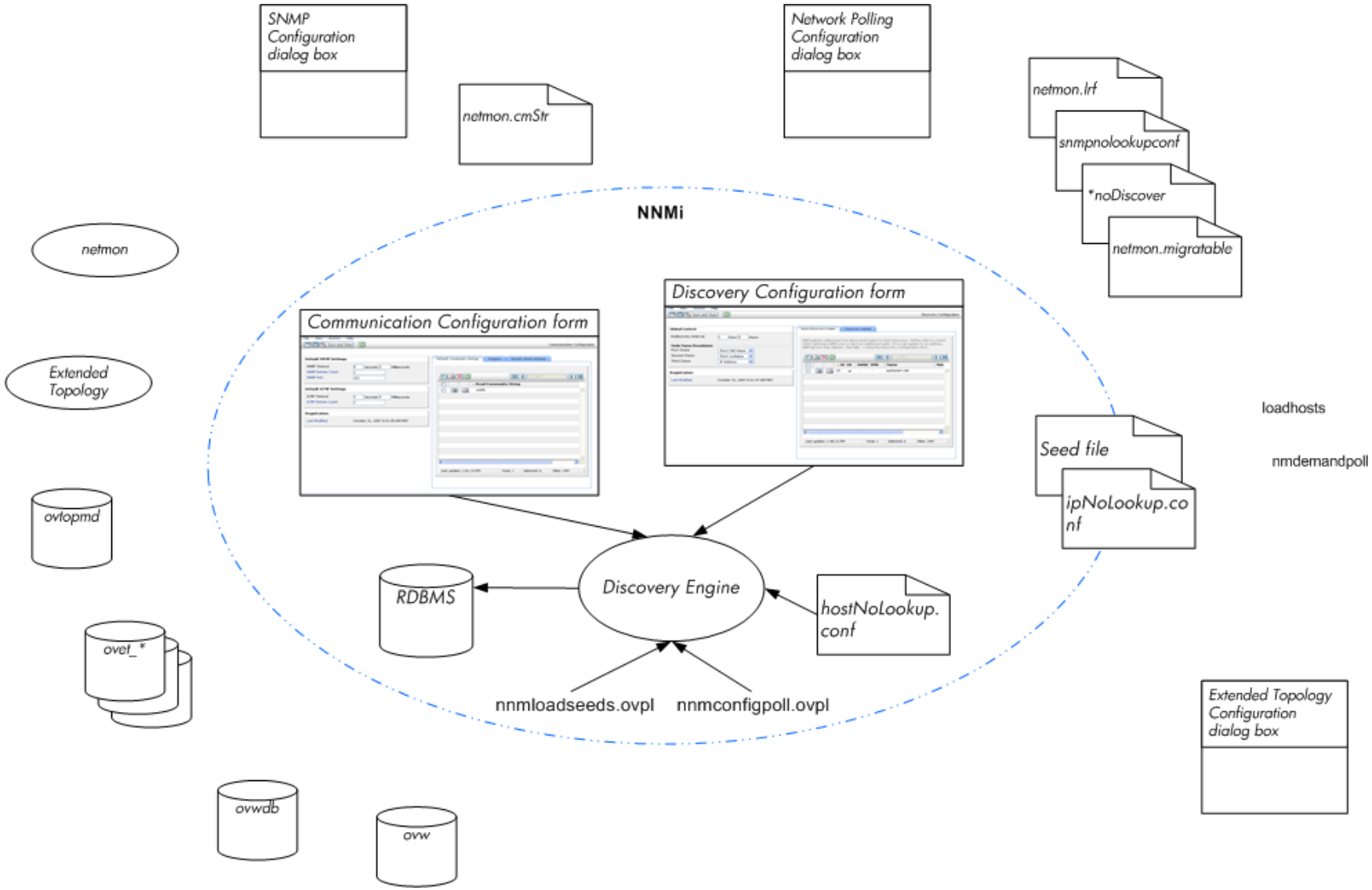
Discovery controls the network elements (devices, nodes, and their components) that are added into the database. In NNMi, “inventory discovery” refers to the activity of finding new nodes and “Layer 2 discovery” refers to the connectivity modeling previously performed by Extended Topology discovery.

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

The center of Figure 8 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 8 Discovery Configuration Elements**



## Key Concepts for Discovery

This section briefly describes the main areas of change from NNM 6.x/7.x to NNMi. For more information about NNMi discovery, refer to *Discovering Your Network* in the NNMi help.

- NNMi stores all information in one relational database.
- NNMi uses one consolidated discovery engine that is easy to configure.
- The NNMi Spiral Discovery process provides ongoing updates to the topology information as changes occur in your network. Topology changes (both inventory and Layer 2) can be discovered more frequently than the scheduled rediscovery interval.
- In NNMi, all discovered nodes are counted against the license limit regardless of the management mode (MANAGED, UNMANAGED, or OUT OF SERVICE). You cannot discover nodes beyond the license limit.
- Auto-discovery has the same meaning in NNMi as in NNM 6.x/7.x, but the configuration approach is different.
  - In NNMi, you define the auto-discovery boundaries, provide at least one IP address seed, and then let discovery run.
  - NNMi auto-discovery uses an expanding model that is easy to control. NNMi auto-discovery finds and manages all routers, switches, and subnets within the boundaries that you provide. You specify the additional device types that NNMi should discover and manage.

    ➤ By default, non-SNMP nodes are *not* discovered in NNMi.
- Seeded discovery has the same meaning in NNMi as in NNM 6.x/7.x, but the configuration approach is different.
  - In NNMi, you can specify discovery seeds in the user interface.
  - You can use your NNM 6.x/7.x seed files in NNMi without modification.
  - The NNMi `nnmloadseeds.ovpl` command replaces the NNM 6.x/7.x `loadhosts` command.
- The NNMi configuration poll (`nnmconfigpoll.ovpl`) replaces the NNM 6.x/7.x demand poll (`nmdemandpoll`) for determining device configuration information.

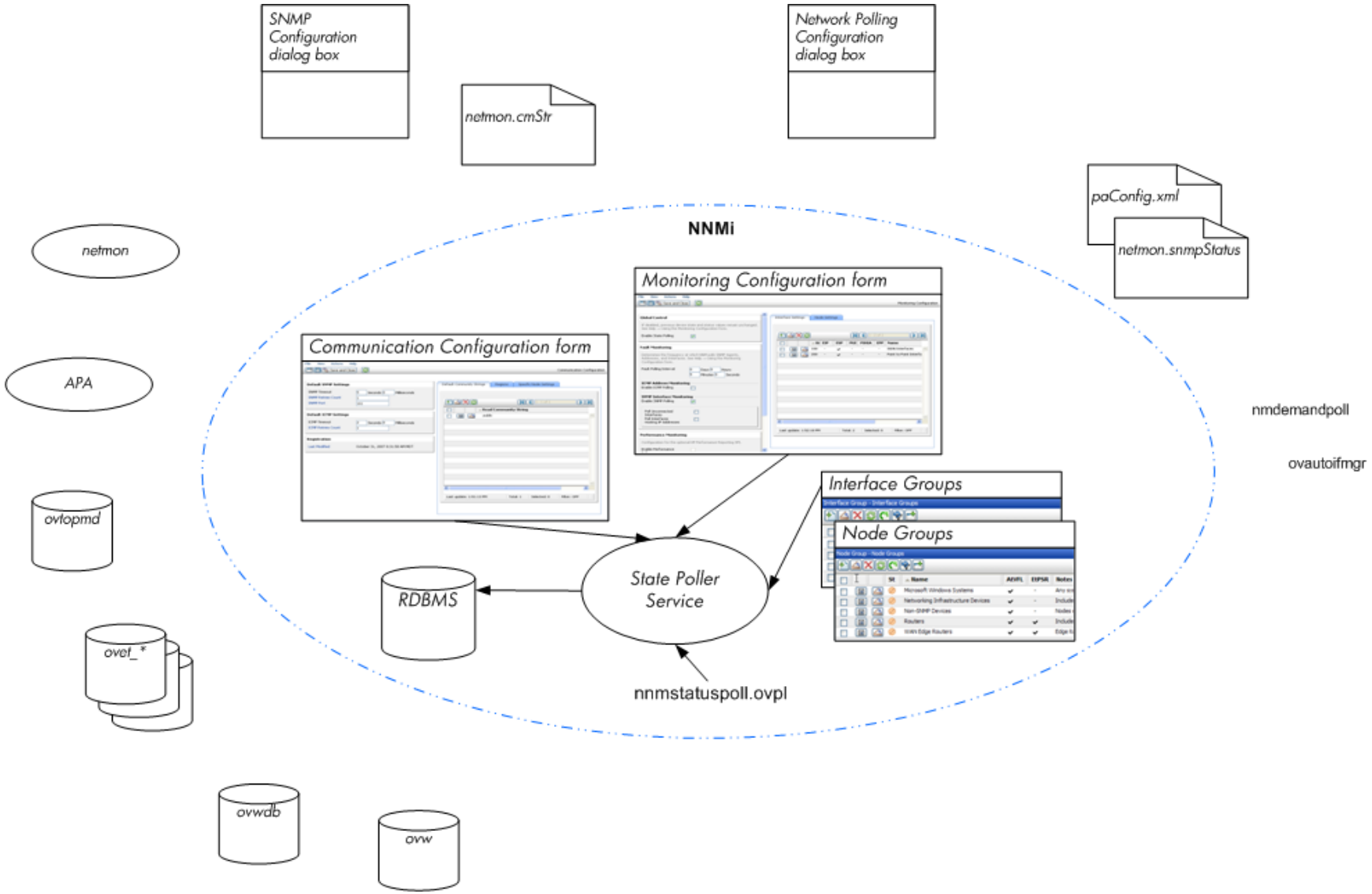
# Status Monitoring

Status monitoring ensures that your network visualization is up-to-date in terms of devices or components that may have faults. When an element fails a poll, NNMi investigates the cause and issues a root cause alarm to the Incident Browser.

The center of Figure 9 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 9 Monitoring Configuration Elements**



## Key Concepts for Status Monitoring

This section briefly describes the main areas of change from NNM 6.x/7.x to NNMi. For more information about NNMi status monitoring, refer to *Monitoring Network Health* in the NNMi help.

- Configuration is now completed through the user interface.
- NNMi node groups and interface groups replace topology filters.
  - Groups can be filtered on a pre-defined set of attributes only.
  - Groups cannot be connected with Boolean operators.
  - Node groups use device filters instead of relying on sysObjectId wildcards.
  - Interface groups can be restricted based on the group of the containing node and the interface type.
- ICMP polling is disabled by default.
- Broad controls make it easier to exclude uninteresting interfaces.
- Monitoring settings are matched from most specific to most general: (1) interface settings, (2) node settings, (3) defaults.
- To change a monitoring behavior across the system, change all settings at all levels.
- The NNMi status poll (**Actions > Status Poll** or `nnmstatuspoll.ovpl`) replaces the NNM 6.x/7.x demand poll (`nmdemandpoll`) for determining device status.
- By default, NNMi only polls interfaces that are connected to another known interface through a Layer 2 connection. You can enable polling of unconnected interfaces and of interfaces that host IP addresses.

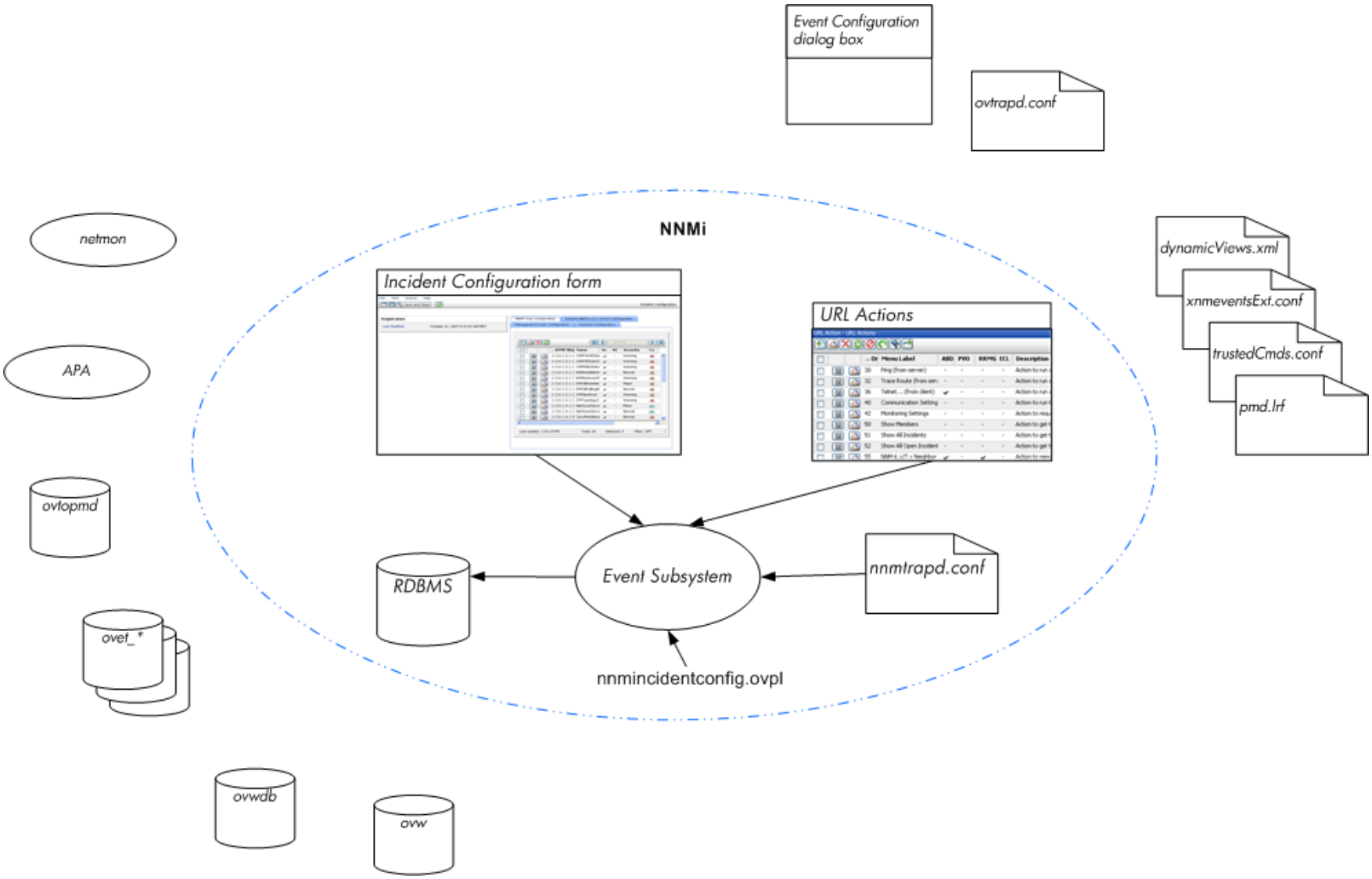
# Customizing Event Monitoring

NNMi provides one centralized location, the incident views, where the management events, SNMP traps, and NNM 6.x/7.x forwarded events are visible to your team. You control which SNMP traps and NNM 6.x/7.x events are considered important enough to appear as incidents.

The center of Figure 10 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 10 Event Monitoring Configuration Elements**

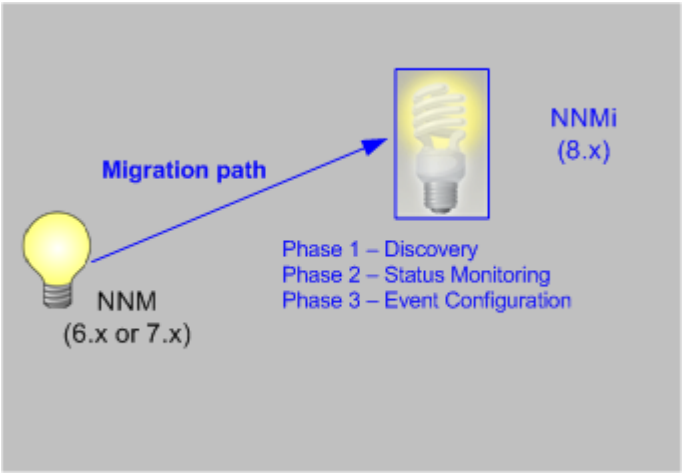


## Key Concepts for Event Monitoring

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

- In NNMi, the event subsystem is not used for inter-process communication and the volume of events is significantly reduced. The administrator no longer needs to configure whether each IPC message should be displayed or logged.
- NNMi receives only those traps for which it is configured. Unconfigured traps are filtered out of the event pipeline.
- NNMi displays all traps that it receives.
- Trap filters for the NNMi event subsystem processes are configured implicitly based on the selections in the **Incident Configuration** form.
- The NNMi `nnmincidentconfig.ovpl` command loads only the trap definitions from the named MIB files.
- NNMi provides pairwise, rate, and de-duplication correlations that occur in the event pipeline. (NNMi does not include the event correlation system (ECS).)
- In NNMi, you can configure actions that occur at any point in the lifecycle of an incident. Actions can be any script, executable, or Jython action.
- The NNMi URL Actions configuration replaces the `dynamicViews.xml` and `xnmeventsExt.conf` configuration files for defining actions that can be taken as a result of an event.

# Migrating from NNM 6.x/7.x




This chapter provides a basic migration path from HP Network Node Manager 6.x or 7.x to HP Network Node Manager i Software. This path should serve the needs of most users. This chapter does not cover advanced migration topics or customizations; consulting services are available to meet your needs in these areas.

This chapter uses the following product naming conventions:

- **NNM** refers to older versions of HP Network Node Manager (including all 6.x and 7.x releases of NNM).
- **NNMi** refers to HP Network Node Manager i Software (including all 8.x releases of NNMi and NNMi Advanced).

This chapter makes the following assumptions:

- You have installed NNMi following the instructions in the *NNMi Installation Guide*.
- You have reviewed the concepts described in the NNMi help and the deployment information in this guide and for a general understanding of NNMi functions.
- You understand how to use the NNMi console.

The information in this chapter will be updated frequently as migration tools are released and as NNMi evolves.

For up-to-date, downloadable copies of NNM and NNMi documentation, go to:

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

This chapter contains the following topics:


- Migration Options
- Phase 1: Migrate SNMP Information
- Phase 2: Migrate Discovery
- Phase 3: Migrate Status Monitoring
- Phase 4: Migrate Event Configuration and Event Reduction

# Migration 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 and Migration Guide*. You do not need to read this migration chapter.

## Phased Migration

For some organizations, a phased approach to migration works better than a new installation. These organizations require that the new NNMi implementation completely reproduce and replace the existing NNM implementation. While there are many possible paths to that end, HP recommends the following phases:

- Phase 1: Migrate SNMP Information

  Configure NNMi with the SNMP access information for your environment.

- Phase 2: Migrate Discovery

  Configure NNMi to discover the objects that were discovered by NNM by approximating the way that NNM discovered them (automatically).

- Phase 3: Migrate Status Monitoring

  Configure the status polling intervals and protocols that are most appropriate for your environment.

- Phase 4: Migrate 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 5: Migrate Graphical Visualization

  Migration of NNM `ovw` customizations and container views containers is not available at this time.

NNMi can act as a manager of managers for your existing NNM systems. You can configure NNM to forward events to NNMi. Then you can use the NNMi console, with its consolidated user interface, incident ownership, and lifecycle states to navigate to familiar NNM tools. For instructions on integrating NNM into NNMi, see Integrating NNM 6.x or NNM 7.x with NNMi 8.10 on page 173.

Table 15 presents a high-level overview of the migration process for the two ends of the migration complexity continuum:

- The simplest approach involves importing environment-specific information from NNM and accepting the default NNMi configuration values, which are improved from NNM.
- The most detailed and thorough approach takes a close look at the NNM configuration and replicates this configuration in NNMi.

The remainder of this chapter walks through the process of migrating an NNM configuration to NNMi. The text in the left margin indicates how the specific steps fit into the migration process:

- **Gather from NNM** indicates work to be done on the NNM management station.
- **Replicate to NNMi** indicates work to be done on the NNMi management server.
- **Enhance in NNMi** indicates optional work to be done on the NNMi management server. You can perform enhancements during the migration process or at any time in the future.

At appropriate points, you will be given two or more options along the complexity continuum for completing a given task.

**Table 15 Migration Continuum**

| Phase | Simplest Approach | Most Detailed and Thorough Approach |
|---|---|---|
| **SNMP Information** | 1 Export all community strings currently in use.<br>2 Import these community strings into NNMi and let NNMi sort out which community string goes with which node. | 1 Export all community strings currently in use.<br>2 Modify the data file and import the contents to NNMi as specific node community strings. |
| **Discovery** | 1 Export a list of all discovered nodes.<br>2 Modify the data file 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).<br>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.<br>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.<br>2 Add the definitions for any custom traps from managed devices.<br>3 Add automatic actions as necessary. | 1 Determine exactly what NNM customizations have been made for each trap and event type.<br>2 Customize each matching trap and event type on the NNMi system. |

# Phase 1: Migrate SNMP Information

Migrate the SNMP community string information that NNMi uses to establish connections with managed devices.

If the NNM configuration includes IP addresses or hostnames that should not be looked up in the name resolution service, migrate that information to NNMi.

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

## Configure SNMP Access

NNMi discovery requires SNMP access to the managed nodes in order to collect specific information about their configuration and connectivity. SNMP is also used during status monitoring to assess the health of the node and the objects it contains.

NNM tries community strings serially, in the order listed for the matched region, and uses the first one that works. NNMi tries all configured community strings in parallel and uses the first one that works. Use the best community string where there might be multiple working values.

**Gather from NNM**

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.mycorp.net:public:*::::::
127.0.0.1:public:*::::::
10.97.233.209:mycommstr:*::::::
mpls2950.mycorp.net:mycommstr:*::::::
mplsce04.mycorp.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.

2. Review any configured alternative community strings in the following file:
   - *Windows*: `%OV_CONF%\netmon.cmstr`
   - *UNIX*: `$OV_CONF/netmon.cmstr`

## Replicate to NNMi

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 1 on page 148.

The following configuration areas are not transferable: `SNMP proxy system` and `Set community name`.

### Simple approach

The easiest approach is to enter all NNM community strings and let NNMi determine the SNMP community string to use for each device. Community string discovery is enabled by default; you can use this feature to expedite migration.

1. Notify your network operations center (NOC) to expect authentication errors during NNMi's initial discovery. NOC personnel can safely ignore these authentication errors during that time.
2. Complete one of the following actions:
   - Modify the `snmpout.txt` 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.

      — *Windows*: Open the `snmpout.txt` file in Microsoft Office Excel. Select the data rows, and then sort on column B.

      For this example, consider two unique community strings:

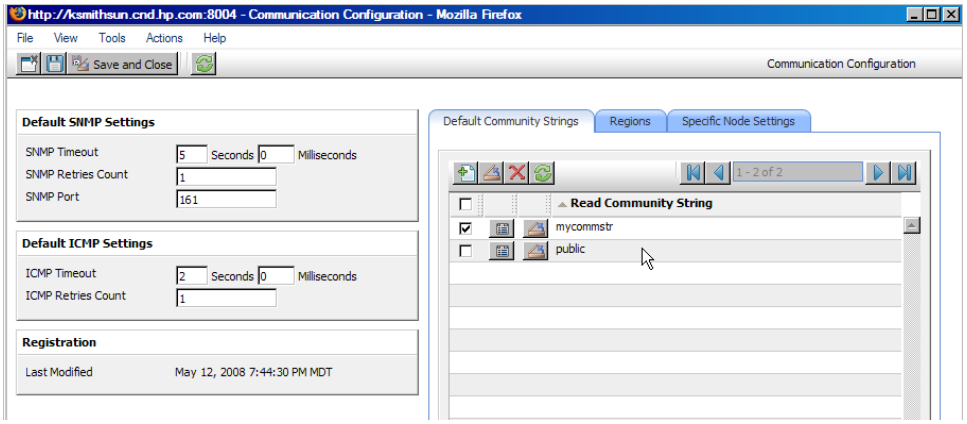
      ```
      public
      mycommstr
      ```

      — *UNIX*: Run the following command:

      ```
      cut -f 2 -d ':' < snmpout.txt | sort -u
      ```

   b. In the NNMi console, select **Communication Configuration** from the **Configuration** workspace. Enter all of the unique values on the **Default** tab.

   c. Configure timeouts, retries, and port.



### Modified simple 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 timeouts, retries, and port.

### Automated 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.mycorp.net,public
     127.0.0.1,public
     10.97.233.209,mycommstr
     mpls2950.mycorp.net,mycommstr
     mplsce04.mycorp.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 timeouts, 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 8.10 also selects the highest 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 Configure SNMP Access on page 148.

**Gather from NNM**

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`
2. Run the following command to determine the **hostnames** that NNM excludes from name-to-address resolution:

```
snmpnolookupconf -dumpCache > snmpnolookup.out
```

**Replicate to NNMi**

3. Add the addresses from step 1 to the following file:
   - *Windows*: `%NnmDataDir%\conf\ipNoLookup.conf`
   - *UNIX*: `/$NnmDataDir/shared/nnm/conf/ipNoLookup.conf`
4. Add the hostnames from step 2 to the following file:
   - *Windows*: `%NnmDataDir%\conf\hostNoLookup.conf`
   - *UNIX*: `$NnmDataDir/shared/nnm/conf/hostNoLookup.conf`

For information on the format of these configuration files, refer to the *ipNoLookup.conf* and *hostNoLookup.conf* reference pages, or the UNIX manpages.

**Enhance in NNMi**

NNMi does lookups during discovery only. By replicating the NNM no-lookup configuration to NNMi, the spiral discovery operation is automatically enhanced.

5. 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:
   a. In the NNMi console, open the **Configuration** workspace.
   b. Select **Discovery Configuration.**
   c. Set your node name preferences in the **Node Name Resolution** area.

## Customize Device Profiles

NNM collects some configuration information directly from SNMP queries to the device. Other information is *derived* from the device's **system object ID** (`sysObjectID`). Mapping from the `sysObjectID` to attributes in NNMi is done through **device profiles**. Device profiles are used to group nodes for monitoring, to filter nodes for viewing, and to categorize nodes for discovery maintenance.

The following configuration areas are not transferable:

- Custom symbols
- Custom database fields and default values

**Gather from NNM**

1 Determine any customizations to the OID files for your version of NNM.

- NNM 6.4 and earlier used the files `oid_to_sym`, `oid_to_type`, and `HPoid2type` to map a system's `sysObjectID` to database attributes and displayed symbol.
- NNM 7.x replaces the `oid_to_sym` file with the `oid_to_sym_reg` directory structure.

**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.

2 In the NNMi console, select **Device Profile Configuration** 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 2: Migrate Discovery

Migrate the discovery schedule and configuration. NNMi spiral discovery begins immediately after you save one or more discovery seeds.

Configure NNMi to use the appropriate community strings for your network environment before initiating discovery.

After initial discovery, migrate any connections between devices that were configured manually in NNM.

## Schedule Discovery

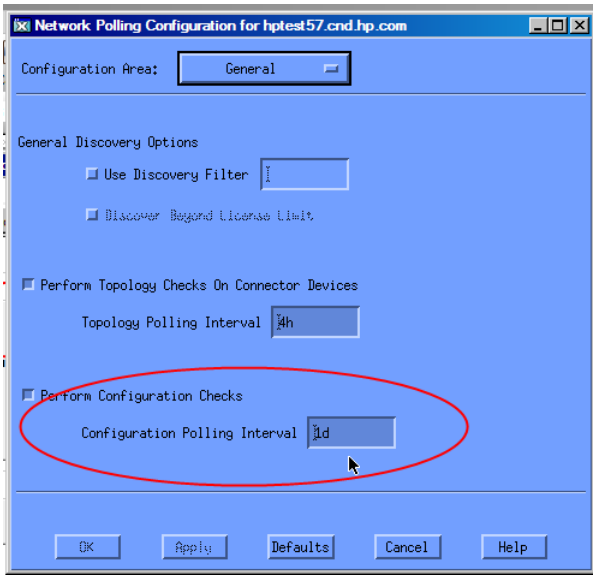
The NNM discovery processes can run independently. To migrate discovery to NNMi, you only transfer the **interval** at which NNM discovers nodes.

The following schedule configuration areas are no longer used in NNMi and are not transferable:

- Topology checks on connector devices. A topology check now happens automatically whenever NNMi sees a trigger that indicates a possible change.
- Configuration check. A configuration check now happens at the time of a scheduled discovery or with any trigger in NNMi.
- Layer 2 (Extended Topology) discovery behavior. NNMi performs Layer 2 discovery for each device as it is found, so there is no need to schedule this behavior separately.
- Auto-adjusting discovery polling interval.

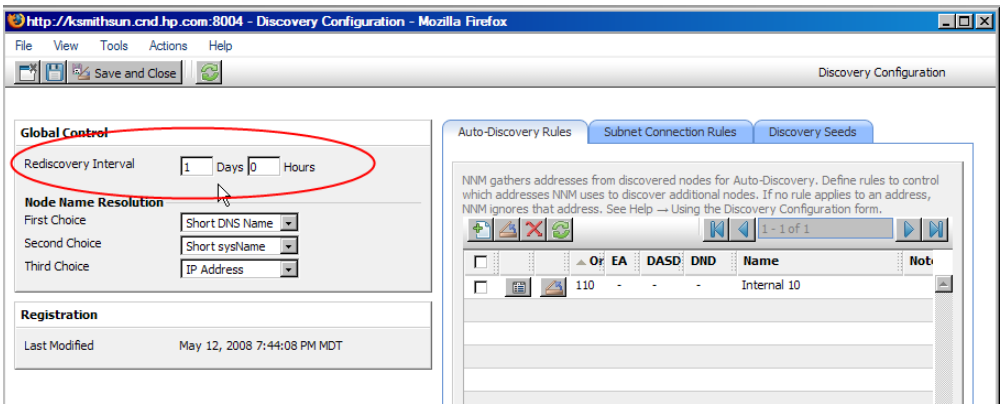
### Gather from NNM

1. Determine when NNM performs rediscovery.
   a. In a user interface, select **Options** → **Network Polling Configuration**.
   b. On the **IP Polling** page, review the **Discovery polling interval** box.
      — If NNM uses a fixed interval, note that value for transfer to NNMi.
      — If NNM uses auto-adjusting intervals, NNM waits a maximum of 24 hours. You can choose to stay with 24 hours, or you can select a new value.
      — If auto-discovery has not been not enabled, determine the interval for **Perform configuration checks** on the **General** page and note that value for transfer to NNMi.



**Replicate to NNMi**

2 In the NNMi console, select **Discovery Configuration** from the **Configuration** workspace, and then set the **Rediscovery Interval** to the value determined in step 1.




**Enhance in NNMi**

All other configuration updates are automatic and incremental, so configuration is simpler and discovery is more efficient than in NNM.

## Select Your Discovery Method

Determine which model to use for NNMi discovery:

- Seeded discovery with no auto-discovery rules. This type of discovery is bounded by the administrator, who controls what is discovered by adding seeds as necessary. Complete *only* the following task:
  - Add Seeds to NNMi for Seeded Discovery on page 159
- Automatic discovery based on seeds and auto-discovery rules. Complete both of the following tasks:
  - Configure Auto-Discovery Rules on page 155
  - Add Seeds to NNMi for Seeded Discovery on page 159

For more information on the differences between the NNMi discovery methods, refer to *Determine Your Approach to Discovery* in the NNMi help.

NNM licenses are based on the number of nodes under management (status monitoring). NNMi licenses are based on the number of nodes discovered and placed in the topology (monitored and unmonitored nodes).

While this difference might encourage you to discover fewer nodes, there are advantages to including unmonitored nodes in your database. For example:

- You might want to see a service provider's access router and your connectivity to it, even if you are not responsible for managing the device.
- Status monitoring algorithms are based on connectivity as seen in the database. Interfaces having no device on the other end of the link *in the database* are unmonitored by default. You might choose to override the default in status monitoring configuration, or you might choose to discover the device. Your choice depends on the balance of interests in your environment. For more information, see Interfaces to Unmonitored Nodes on page 60.

## Configure Auto-Discovery Rules

NNMi discovery configuration provides an excellent opportunity to consider what you want to manage with NNMi. Before you invest in converting your NNM discovery configuration and filters, consider looking at your current network environment and describing what you want to include in the NNMi topology.

If you do want to invest in direct conversion, NNMi discovery rules encompass two task sets from NNM: extending the scope of discovery and limiting the objects discovered within that scope.

For NNMi configuration, it is important to define all of the rules to extend and/or limit discovery before entering the seeds, which initiates the discovery process.

The following schedule configuration areas are no longer used in NNMi and are not transferable:

- IPX discovery from Windows
- Discover beyond license limit
- Disable discovery of Layer 2 objects (always enabled for NNMi)
- `netmon.interfaceNoDiscover`
- Discovery exclusions by filtering on attributes other than IP address and `sysObjectID` (and its derivatives)
- Limiting Layer 2 discovery through `bridge.noDiscover`
- Limiting Layer 2 discovery based on CDP protocol area (aggregated ports, vlans, etc.)
- Extended Topology zone configuration, which is no longer relevant to NNMi's spiral discovery

## Configure Spiral Discovery

NNMi provides two methods for configuring spiral discovery in NNMi: manually loading nodes (for example, from a host file) and using auto-discovery rules.

### Load nodes manually

**Gather from NNM**

1. In NNM, find the file that contains the output of the `loadhosts` command. This file lists an IP address and a hostname for each node, plus a subnet mask if one was specified.

   NNM loadhosts example

   An example file for the `loadhosts` command looks similar to the following:

```
10.2.32.201 lnt04.mycorp.net  # comment
10.2.32.202 lnt07.mycorp.net  # comment
10.2.32.203 lnt03.mycorp.net  # comment
10.2.32.204 lnt02.mycorp.net
10.2.32.205 lnt05.mycorp.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.

Complete all community string configuration prior to configuring any seeds into NNMi.

If you want the discovery output to be equivalent to NNM `loadhosts`, disable any auto-discovery rules that are configured in NNMi. To disable an auto-discovery rule, do one of the following:

- Delete the rule from the **Discovery Configuration** form.
- On the **Auto-Discovery Rule** form, clear the **Discover Included Nodes** check box.

The format for the seed file in NNMi is either an IP address or a node name (plus an optional comment) per line. For more information, refer to the *nnmloadseeds.ovpl* reference page, or the UNIX manpage.

NNMi seed file example

The following example shows an NNMi seed file with the same function as the NNM `loadhosts` command and a hostfile:

```
10.2.32.201 # comment
10.2.32.202 # comment
lnt03.mycorp.net  # comment
lnt02.mycorp.net
10.2.32.205
```

Best practice

The following file contains a list of devices from Extended Topology:

- *Windows*: `%OV_DB%\nnmet\hosts.nnm`
- *UNIX*: `$OV_DB/nnmet/hosts.nnm`

You can copy the first field (IP address) or second field (nodename) to create a seedfile for NNMi.

On UNIX, you can run the following command to create a file of the node names:

```
cut -f 2 hosts.nnm
```

Best practice

NNMi always favors the loopback address as the management address. If you do not use loopback addresses, NNMi probably (but not always) uses the seed address as the management address. Therefore, it is a good practice to populate the hostfile with preferred IP addresses. If you use hostnames, verify that the DNS resolves to the preferred management address, which still does not guarantee that NNMi will use this address as the management address. For more information about management address selection, refer to *Discovery Node Name Choices* in the NNMi help.

### Use auto-discovery rules

Gather from NNM

1. Determine whether a discovery filter was used for NNM. In NNM, one discovery filter applied to the entire scope of discovery.
   a. Open an NNM user interface.
   b. Select **Options** → **Network Polling Configuration**.
   c. On the **General** page, review the **Use filter** check box and, if selected, note the discovery filter in use. If no filter is in use, continue with Add Seeds to NNMi for Seeded Discovery on page 159.
   d. Locate the discovery filter in the following file:
      — *Windows*: `%OV_CONF%\C\filters`
      — *UNIX*: `$OV_CONF/C/filters`

      Review the 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 }
```

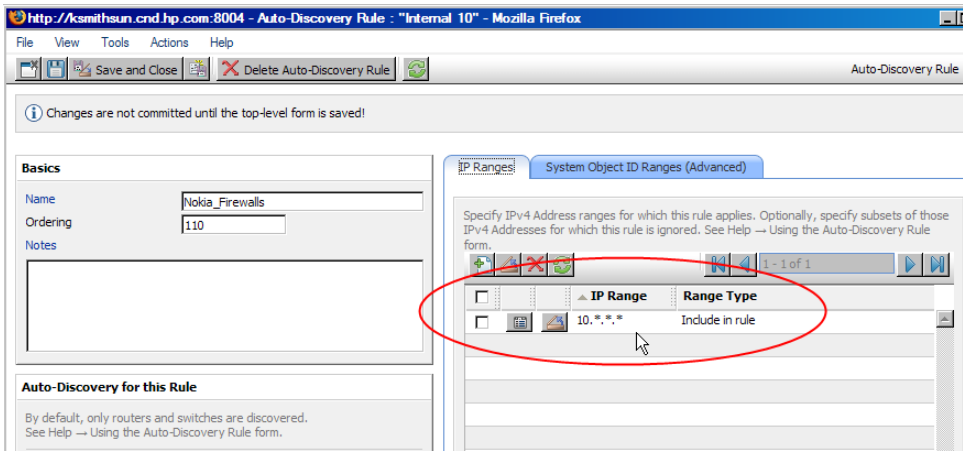
## Replicate in NNMi

2 Enter the discovery filters in the NNMi console.

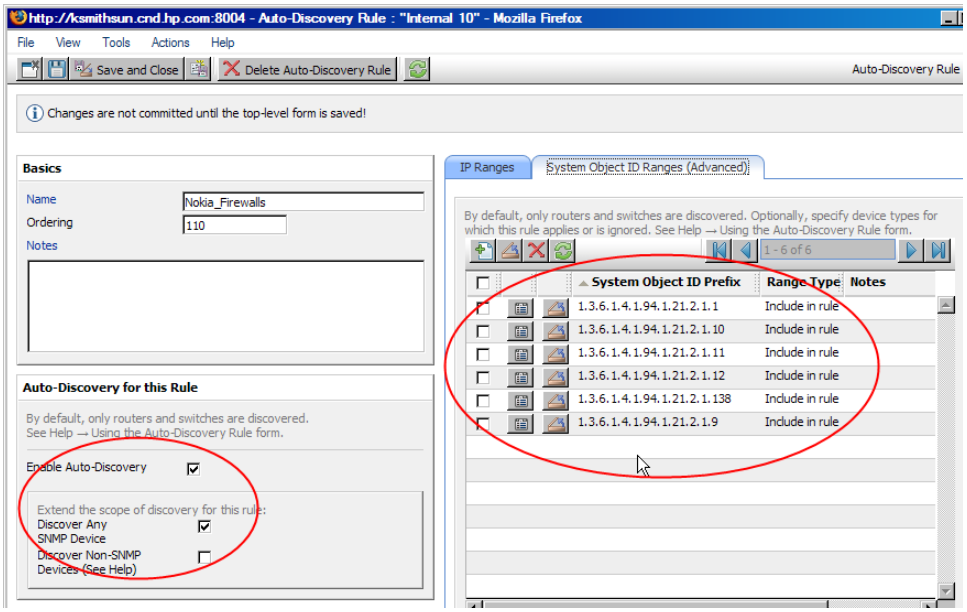
NNMi discovery filter entry example

For example, to transfer the NNM filter shown in the NNM discovery filter example on page 157 to NNMi, you would define three auto-discovery rules: one rule for Nokia firewalls, one rule for NetBotz devices, and a final rule for Routers and Switches (same as Bridge in NNM 7.x). NNMi does not require NetsNSegs. For this example, assume that the range of the network to be discovered is 10.*.*.*.

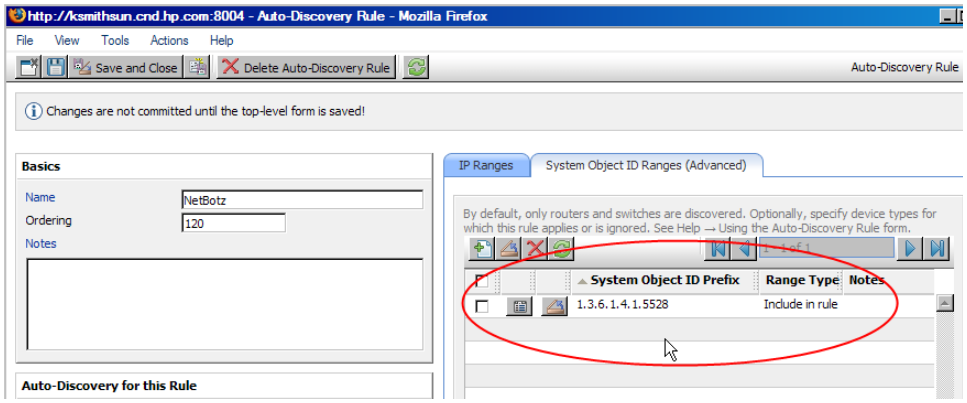
a For Nokia firewalls, enter a rule name (Nokia_Firewalls), and then enter the network IP range 10.*.*.*.



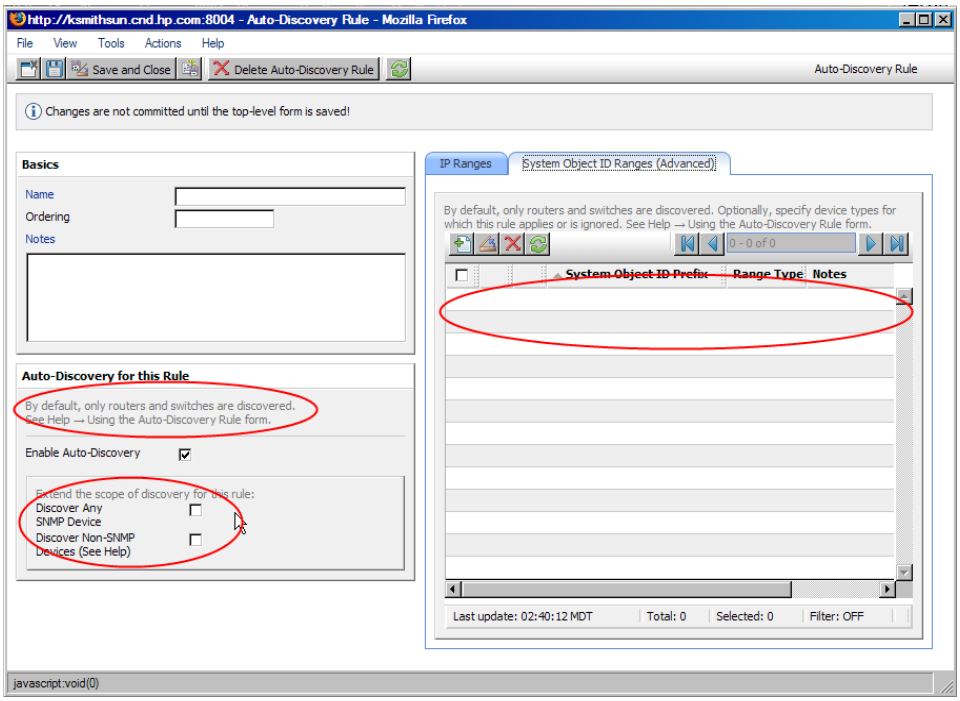
b Enter each `sysObjectID` (do not enter the leading period), and then select the **Discover Any SNMP Device** check box. (By default, NNMi only discovers switches and routers. Because these devices might not be marked as switches or routers, select the **Discover Any SNMP Device** check box when specifying `sysObjectIDs`.)




c Enter the NetBotz rule. This rule uses a wildcard in NNM: `.1.3.6.1.4.1.5528.*`. In NNMi the asterisk (.*) is implied and not required.



d The final rule is for switches and routers. Because NNMi discovers these devices by default, do not specify Object IDs (OIDs). You only need to specify the IP Range.




## Add Seeds to NNMi for Seeded Discovery

**Gather from NNM**

1 Determine the exact list of devices in the NNM database by running the following command:

```
ovtopodump > topology.out
```

**Replicate in NNMi**

2 Copy and edit the `topology.out` (export) 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 mask because NNMi determines the subnet mask automatically.

```
10.2.32.201 # comment
10.2.32.202 # comment
lnt03.mycorp.net  # comment
lnt02.mycorp.net
10.2.32.205
```

Alternatively, you can add this list of nodes by using the NNMi console.

3 Run the following command:

```
nnmloadseeds.ovpl -f newSeedfile
```

For more information, refer to the *nnmloadseeds.ovpl* reference page, or the UNIX manpage.

NNMi begins to discover the devices associated with these seeds immediately and implements the existing device profiles (and node groups, such as node groups for status monitoring). NNMi spiral discovery is ongoing. For information on how to determine discovery status, refer to *Check Discovery Progress* in the *NNMi Installation Guide*.

## Customize Connectivity

In certain circumstances where device information is limited, NNM's Extended Topology might not accurately discover and model every connection in a network. As a result, you might see no connections where you know connections exist or connections indicated where you know none exist. The remedy for this situation is to create the correct connections manually.

**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.mycorp.net[ifAlias:MyAlias],N2.mycorp.net[ifAlias:MyOtherAlias]
Y1.mycorp.net[ 0 [ 999 ]],Y2.mycorp.net[ 0 [ 2 ]]
```

**Replicate to NNMi**

2 Use the `nnmconnedit.ovpl` tool to make connection edits in NNMi. The file format is completely different from that used by NNM.

a Generate a connection template file by running the following command:

```
nnmconnedit.ovpl -t addconn
```

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

b Edit the template file (`addconn.xml`) to change or add connections. Use the documentation in the file for the syntax of the new file.

**NNMi connection example**

The following example shows the NNMi equivalent to the NNM 7.x example in step 1:

```
<connectionedits>
        <connection>
                <operation>add</operation>
                <node>N1.mycorp.net</node>
                <interface>MyAlias</interface>
                <node>N2.mycorp.net</node>
                <interface>MyOtherAlias</interface>
        </connection>
        <connection>
                <operation>add</operation>
                <node>Y1.mycorp.net</node>
                <interface>999</interface>
                <node>Y2.mycorp.net</node>
                <interface>2</interface>
        </connection>
</connectionedits>
```

c Load the new connection information into the database by running the following command:

```
nnmconnedit.ovpl -f addconn.xml
```

d In the NNMi console, select **Layer 2 Connections** from the **Inventory** workspace to verify the results.

# Phase 3: Migrate 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:

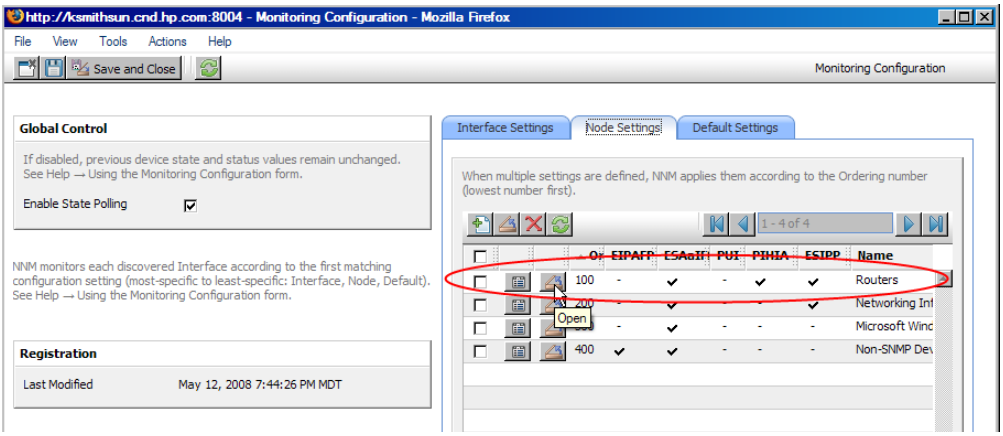
```
<classSpecification>
   <filterName>isRouter</filterName>
   <parameterList>
      <parameter>
         <name>interval</name>
         <title>Interval to Poll Device</title>
         <description>
            The interval for which the device will be polled
               in seconds.
         </description>
         <varValue>
             <varType>Integer</varType>
             <value>300</value>
         </varValue>
      </parameter>
           . . .
   </parameterList>
</classSpecification>
```

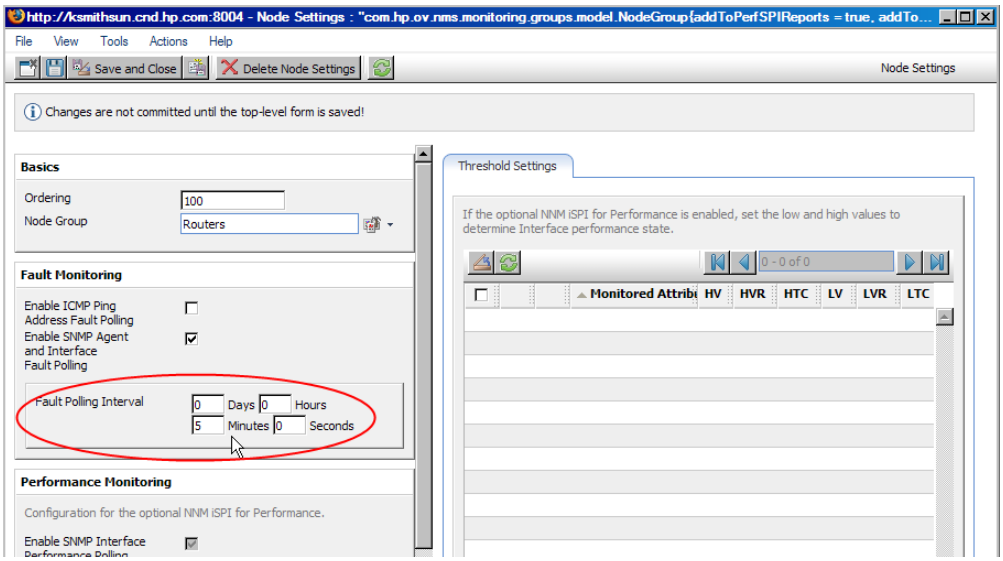
**Replicate to NNMi**

### NNMi polling process

NNMi status monitoring configuration is based on groups of nodes and/or groups of interfaces.

In the NNMi console, select **Monitoring Configuration** from the **Configuration** workspace. Select the **Node Settings** tab, and then set the **Fault Polling Interval** for the group.




## Select Polling Protocol

**Gather from NNM**

### NNM `netmon` polling process

By default, the `netmon` process uses ICMP to poll each address (equated with an interface). NNM can be configured so that the `netmon` process uses SNMP rather than ICMP (it never uses both) for some devices. To determine whether some areas are using ICMP, review the following file:

- *Windows*: `%OV_CONF%\netmon.snmpStatus`
- *UNIX*: `$OV_CONF/netmon.snmpStatus`

### NNM APA polling process

APA uses a combination of SNMP and ICMP for polling. In APA, the polling policies are applied to nodes or interfaces, which are grouped by filters. The filters are defined in the `TopoFilters.xml` file. The polling policies are defined in the `paConfig.xml` file.

## Replicate in NNMi

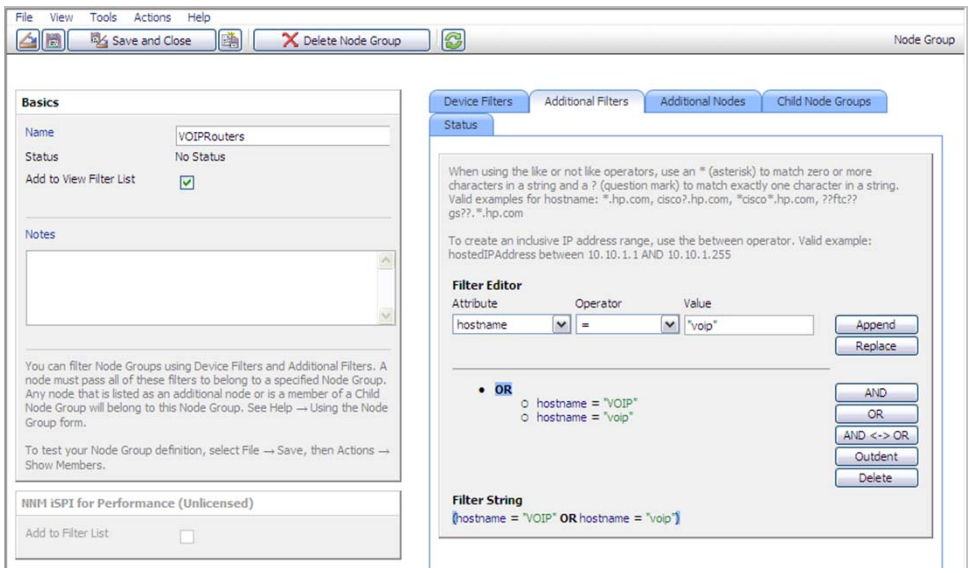
### NNMi polling process

In NNMi, the nodes and interface collections are defined as node groups and interface groups. Polling policies are applied to node groups and interface groups on the **Monitoring Configuration** form.

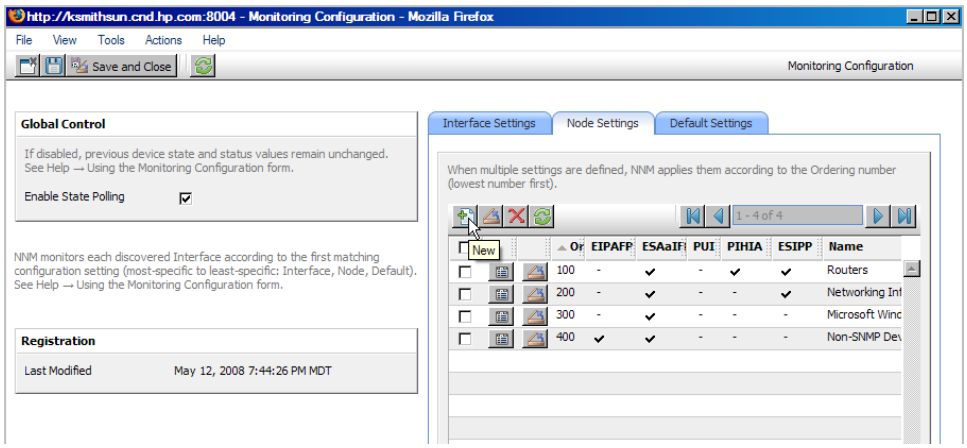
NNMi polling configuration example

For example, to configure polling (using SNMP and ping) for a collection of VOIP routers, follow these steps:

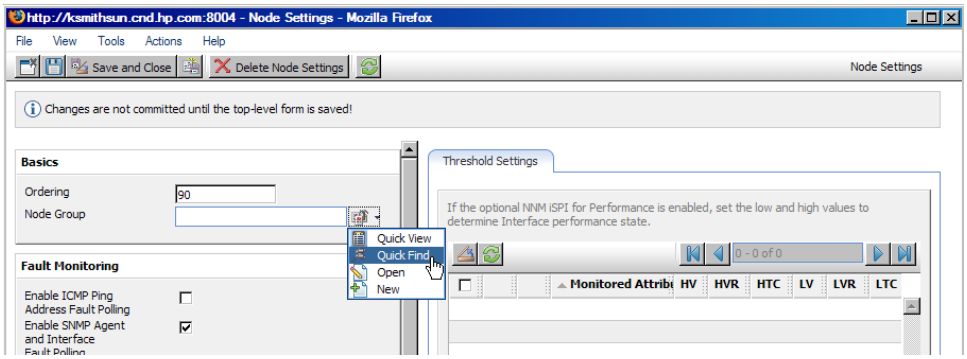
1. Using the **Node Group** form, create a node group that identifies the VOIP routers. Hostname wildcards are case sensitive, as shown here. Save and close this form.



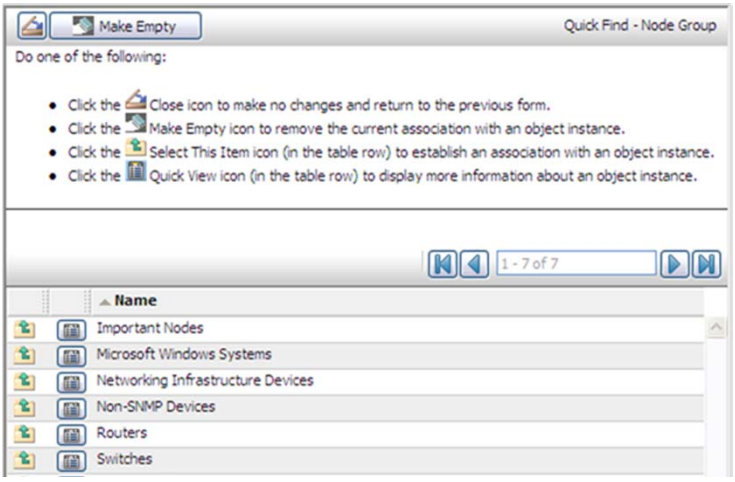

2. On the **Monitoring Configuration** form, add new node settings, as shown here.



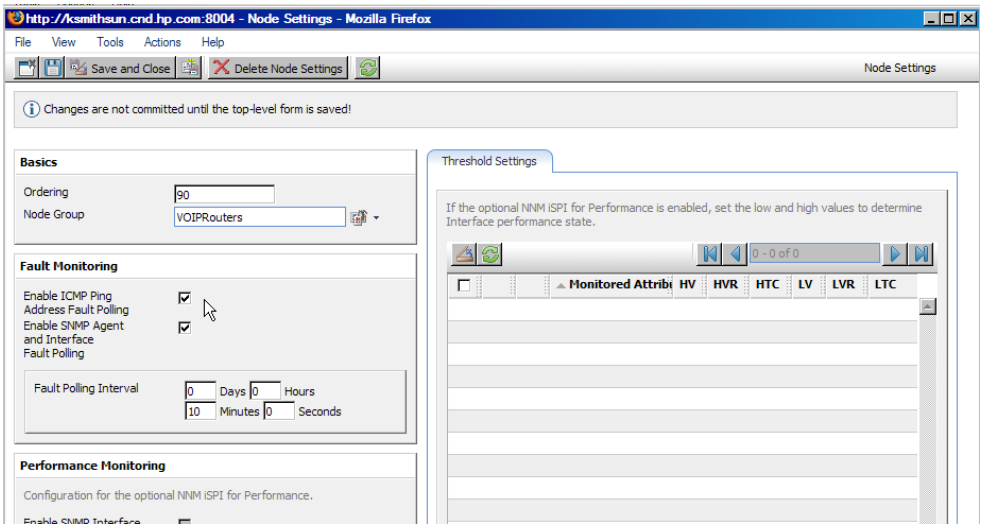
3 Specify an ordering value, and then select quick find for the **Node Group** field, as shown here.




4 Select the node group for these monitoring settings, as shown here.




5 Select the **Enable ICMP Ping Address Fault Polling** check box, as shown here. Save and close the form.



## Configure Critical Nodes

By default, NNMi provides a node group for important nodes. This node group functions in the same way as the critical nodes list in NNM.

When important nodes are down or unreachable, NNMi shows node status as critical and generates a `NodeDown` incident.

### Gather from NNM

#### NNM `netmon` polling process

If NNM uses `netmon` for status monitoring, NNM is not configured for critical nodes. You can create a new critical node configuration in NNMi.

#### NNM APA polling process

Review the following file to determine which nodes are designated as critical for APA:

- *Windows*: `%OV_CONF%\nnmet\topology\filter\CriticalNodes.xml`
- *UNIX*: `$OV_CONF/nnmet/topology/filter/CriticalNodes.xml`

NNM CriticalNode.xml example

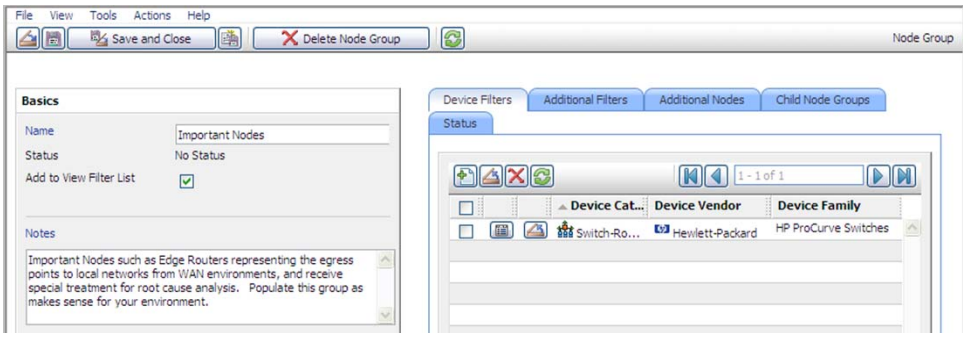
The `CriticalNode.xml` file should resemble the following example:

```
<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.mycorp.net</DNSName>
   <DNSName>router7.mycorp.net</DNSName>
   <DNSName>MPLSRtr*.mycorp.net</DNSName>
</HostIDs>
```

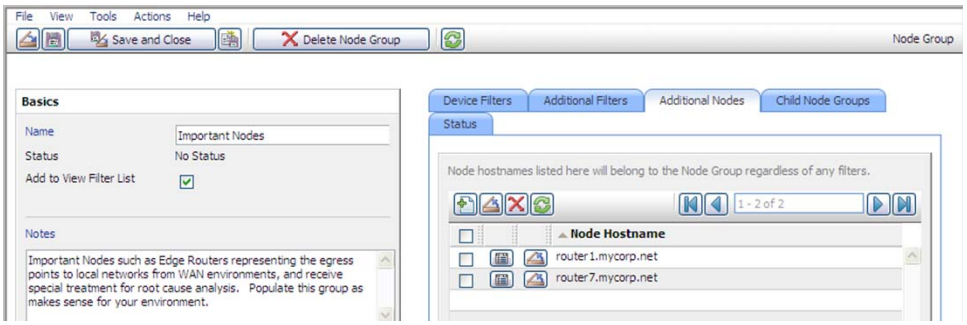
### Replicate to NNMi

#### 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 hostname wildcards.



b Add specific nodes. Save and close the form.

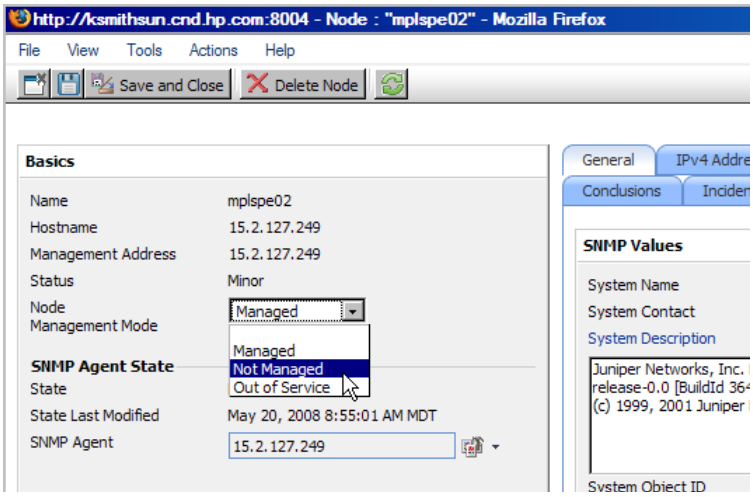



## Exclude Objects from Status Polling

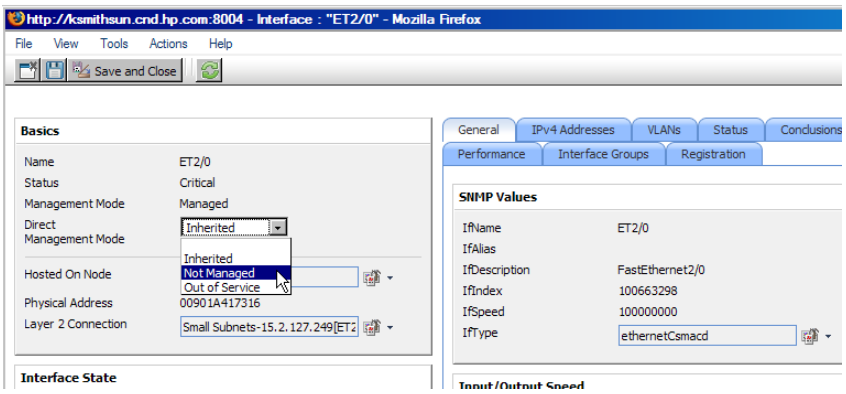
In NNM, most activities that stop nodes or interfaces from being monitored (set them to an UNMANAGED state) are completed through manual intervention in the NNM user interface.

NNMi streamlines the process of unmanaging objects. It is possible that the new product defaults match what you used to do manually (for example, only polling uplinks); however, managing settings through node groups and interface groups makes it easier to update settings automatically.

Occasionally you might need to mark a node or interface as **Not Managed**. You can set the management mode of an individual node on the **Node Details** form, as shown here:




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



# Phase 4: Migrate Event Configuration and Event Reduction

NNM analyzes all sources of incoming events (traps from managed devices, internal process communication, forwarded events) using an extended SNMPv2 format. Each event has an event object identifier, a name, and configuration parameters.

NNMi handles various sources of events differently. Traps from devices and events forwarded from NNM management stations are in the SNMPv2 format. Incidents created within NNMi contain names only; they do not have event object identifiers. In addition, NNMi internal process communications use a new (non-trap) mechanism to significantly improve overall performance. NNMi does not have `no format in trapd.conf` messages for unrecognized events. Unrecognized events are now discarded.

The following event configuration areas are no longer used in NNMi, and are not transferable:

- Event forwarding (can be simulated with an automatic action to call `nnmsnmpnotify`)
- Some Composer correlation types: suppress, enhance, transient, multisource

## Display Traps from Devices

You can configure NNMi to display traps from devices in a way that is similar to the NNM environment.

NNMi contains default configurations for many of the common SNMP and vendor traps shipped with NNM. You can update NNMi with any customizations of these traps.

For a list of variables available for messages and automatic actions, refer to *Configure an Action for an Incident* and *Valid Parameters for Configuring Incident Actions* in the NNMi help.

**Gather from NNM**

1. Determine whether the NNM configuration includes customized traps. Note any customizations made to category, severity, display message, or automatic actions.

**Replicate to NNMi**

2. Download the vendor MIB files to the NNMi management server.
3. Run the following command for each MIB:

   **`nnmincidentcfg.ovpl -loadTraps mibFile`**

   - If one MIB has a dependency on another MIB file, use the following command to preload the dependencies:

     **`nnmincidentcfg.ovpl -loadMib mibFile`**

     Alternatively, you can use the **`nnmloadmib`** command, and then rerun `nnmicidentcfg.ovpl` with the `loadTraps` option.

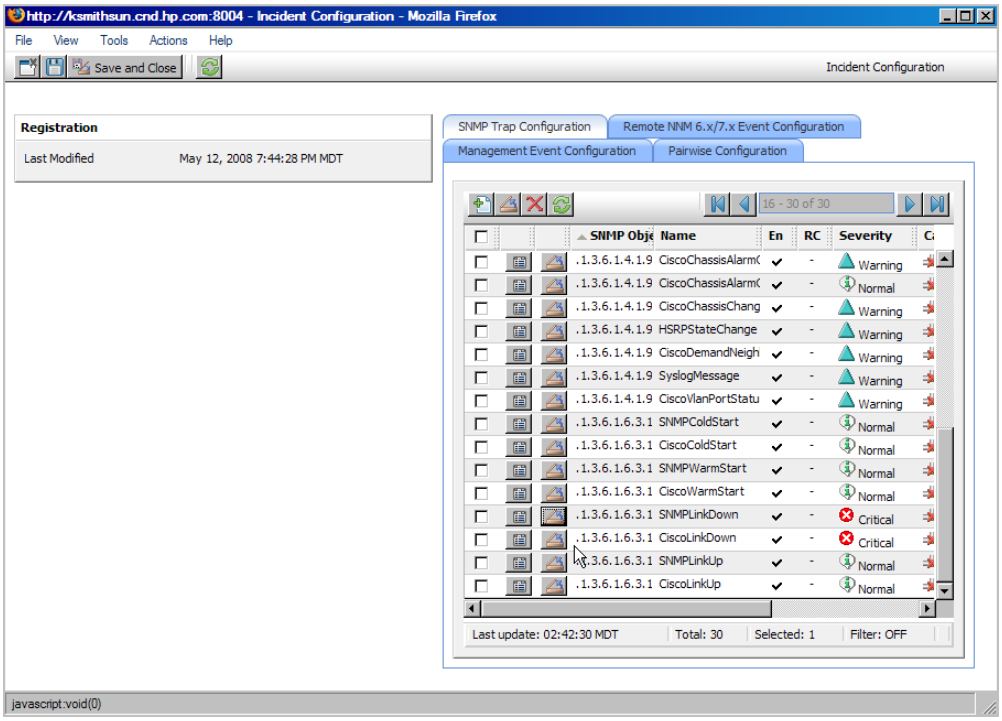
   - To see which MIBs are already loaded, use the command:

     **`nnmloadmib -list`**

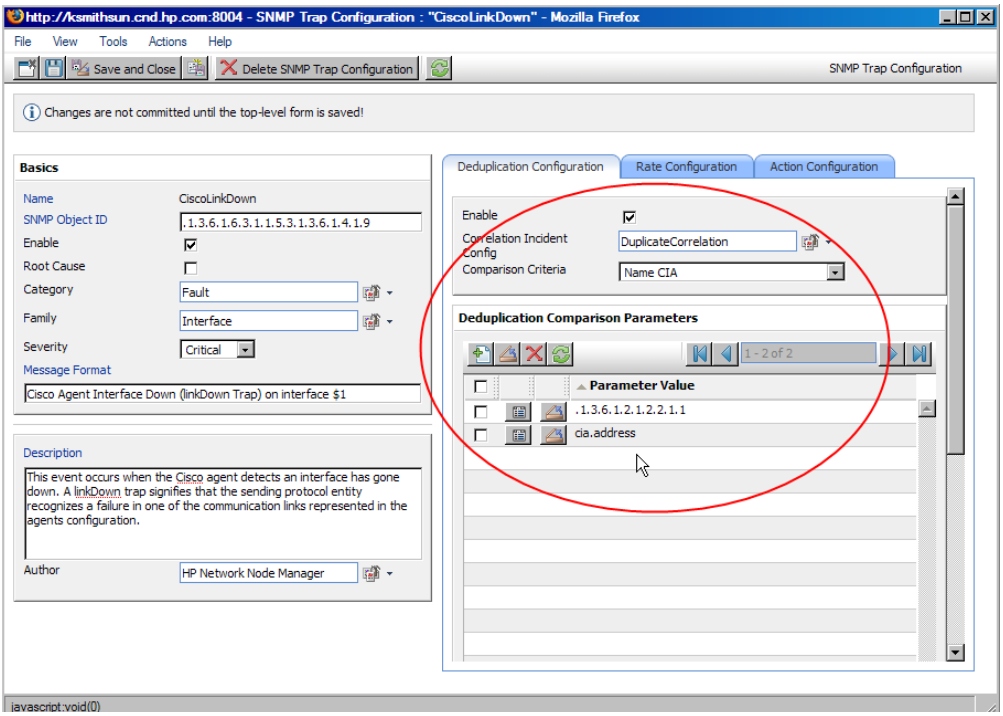
   For more information, refer to the *nnmincidentcfg.ovpl* and *nnmloadmib* reference pages, or the UNIX manpages.

These steps only load TRAP-TYPE and NOTIFICATION-TYPE MIB entries. NNMi does not use other MIB variables.

4 In the NNMi console, select **Incident Configuration** from the **Configuration** workspace. Then select the **SNMP Trap Configuration** tab.




5 Customize the trap displays to match those in NNM. You can create categories as needed on the trap configuration form.




## Enhance in NNMi

6 (Optional) In addition to setting default **Severity**, **Category**, and **Message**, set a default **Family**.

7 (Optional) Classify the trap as a root cause, so that it will appear in the **Root Cause Incidents** view.

## Customize Display of NNMi-Generated Management Events

In NNMi, event configuration is simplified because the NNMi causal engine generates a more concise root cause than NNM.

You can modify the incidents generated with NNMi so that they have a similar appearance to NNM alarms. For example, you can customize the NNMi `NodeDown` incident message to be similar to the message for an NNM `NodeDown` alarm.

**Gather from NNM**

1. In NNM, determine any customizations to the events configuration.

**Replicate to NNMi**

2. In the NNMi console, select **Incident Configuration** from the **Configuration** workspace. Then select the **Management Event Configuration** tab.
3. Locate the new incident configuration by name rather than event number.
4. *Optional*. Customize event displays to match those in NNM by creating categories on the trap configuration form.
5. In addition to setting default **Severity**, **Category**, and **Message**, you can set a default **Family**.

## Block/Ignore/Disable Traps

NNM provides several levels of event processing:

- Block traps as they come into `ovtrapd`
- Process, but do not store or display traps or events labeled `IGNORE`
- Store and process (correlate) events labeled `LOGONLY`, but never display them
- Store, process, and display an event into a category
- Traps that arrive without a configuration appear in the Alarm Browser as `No format in trapd.conf for…` and are stored in the database.

NNMi has a simpler approach. A *disabled* event or trap is not stored, processed, or displayed. An *enabled* event or trap is fully stored, processed, and displayed. Any event for which NNMi does not have a configuration is blocked.

**Gather from NNM**

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**

3. In the NNMi console, select **Incident Configuration** from the **Configuration** workspace. Locate any events that you do not want to receive or display, and clear the **Enable** check box for those events.
4. 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`
5. Use the `nnmtrapconfig.ovpl` command to enable trap blocking and to configure the rates and thresholds for trap blocking.

   For information about using this command, refer to the *nnmtrapconfig.ovpl* reference page, or the UNIX manpage.

## Configure Automatic Actions

**Gather from NNM**

1. Determine any automatic actions that have been configured for NNM.

**Replicate to NNMi**

2. Copy action scripts from the NNM management station to the NNMi management server, where file location is not important.
3. In the NNMi console, select **Incident Configuration** from the **Configuration** workspace.
4. For each NNM event with an automatic action, configure the corresponding NNMi incident with that action (on the **Action Configuration** tab).

   To match the behavior of NNM, set the **Lifecycle State** to **Registered**.

**Enhance in NNMi**

5. Note the following NNMi configuration techniques:
   - You can configure more than one automatic action to occur when an event arrives.
   - You can configure one or more additional actions for each of the other lifecycle states (**In Progress**, **Completed**, **Closed**).
   - You can pass more incident attributes to the command than in NNM.
   - The procedure is simplified because you do not need to register commands in a separate configuration file before NNMi can run them.

## Configure Additional (Manual) Actions

NNM provides operator actions or additional actions that are available from the menu in the Alarms Browser. You might be able to simulate the NNM actions with URL actions that are available from the NNMi console menu.

**Gather from NNM**

1. Determine any custom operator actions that are in NNM.

**Replicate to NNMi**

2. For these customer actions, determine how to transfer them to be available as URLs.
3. In the NNMi console, select **URL Actions** from the **Configuration** workspace.
4. Click **New**.
5. Provide the **Menu Label**, a **Unique Key**, **Ordering**, **Selection Type**, and the **Full URL** for the action.

## Event Correlation: Repeating Events

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

NNMi creates a new parent and displays it when you select **Stream Correlated Incidents** from the **Incident Browsing** workspace. The original events appear in their configured view.

**Gather from NNM**

1. Determine whether the `RepeatedEvents` correlation is in use for NNM.
2. Determine whether the `Repeated` correlator is in use for NNM.
3. Determine whether deduplication is in use (`dedup.conf` file).

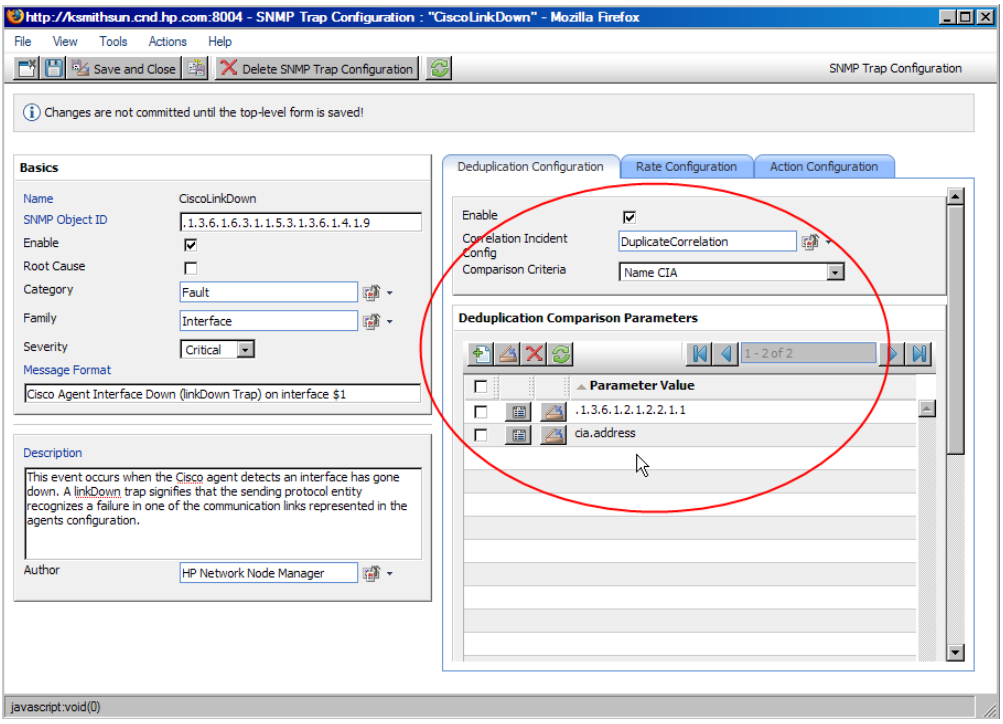
**Replicate to NNMi**

4. In the NNMi console, select **Incident Configuration** from the **Configuration** workspace.

5 Open the event to be deduplicated.

6 Select the **Deduplication Configuration** tab, enable deduplication, select a new parent event, and then define the matching criteria.

Deduplication in NNMi has no time limit.




## Event Correlation: Counting the Rate

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

NNMi creates a new parent and displays it when you select **Stream Correlated Incidents** from the **Incident Browsing** workspace. The original events appear in their configured view, and NNMi has sustained rate behavior equivalent to the rolling time window in NNM.

**Gather from NNM**

1 Determine whether the rate correlator is in use for NNM.

**Replicate to NNMi**

2 In the NNMi console, select **Incident Configuration** from the **Configuration** workspace.

3 Select the **Management Event Configuration** tab.

4 Open the event identifier to be counted.

5 Select the **Rate Configuration** tab, and then do the following:

a Select **Enable** to enable monitoring.

b Set the count window.

c Set the time window (**Hours**, **Minutes**, and **Seconds** fields).

d Select a new parent event (**Correlation Incident Config**).

e Define the **Comparison Criteria**.

For more information, refer to *Management Event Form* in the NNMi help.

## Event Correlation: Pairwise Cancellation

NNMi does not limit cancellation to a specific time window.

Gather from NNM

1 Determine whether the PairWise correlation is in use in NNM.

2 Determine whether the Transient correlator is in use in NNM.

Replicate to NNMi

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

4 Select the **Pairwise Configuration** tab.

5 Select an existing pair, or click **New**.

6 Configure the paired event identifiers and the matching criteria.

For more information, refer to *Configuring Incidents* in the NNMi help.

## Event Correlation: Scheduled Maintenance

NNMi can suppress the monitoring of unavailable nodes. To do so, use the OUT OF SERVICE mode. Unlike NNM, you cannot schedule OUT OF SERVICE maintenance in advance, and you must manually return the objects to MANAGED mode.

SNMP traps sent by devices in OUT OF SERVICE mode are suppressed in NNMi.

If your organization has been using the Scheduled Maintenance correlation, you can use the list of systems that are taken offline together.

Gather from NNM

1 Determine whether the `ScheduledMaintenance` correlation is in use in NNM.

Replicate to NNMi

2 In the NNMi console, select **Node Group Configuration** from the **Configuration** workspace.

3 Create a node group for each set of nodes in the **NNM Maintenance List**. Set the node groups to be available as view filters.

4 When it is time for maintenance, in the NNMi console select **Nodes** from the **Management Mode** workspace.

5 Filter the view to a specific node group by using the **Set node group filter** selector at the top.

6 Select all nodes, and then select **Actions > Out of Service**.

7 After maintenance is completed, select the nodes, and then select **Actions > Manage**.

# Integrating NNM 6.x or NNM 7.x with NNMi 8.10

You can integrate the following HP Network Node Manager (NNM) 6.x/7.x functionality with HP Network Node Manager i Software (NNMi) 8.10:

- You can forward events from NNM 6.x/7.x to the NNMi 8.10 management server in order to use the NNMi 8.10 incident views for managing incident life cycle.
- You can open some NNM 6.x/7.x views from the NNMi 8.10 management server.

This integration is useful for controlling the rate of migration to the new NNMi 8.10 product.

This integration is also useful for large managed environments with many NNM 6.x/7.x management stations. If you do not need the new functionality in NNMi 8.10 throughout the network, you can maintain a few NNM 6.x/7.x management stations while using NNMi 8.10 as your primary network management tool.

You can also use the information in this chapter to integrate a third-party product with NNMi 8.10. That product must be able to generate SNMP v1, v2C, or v3 traps and send them to the NNMi 8.10 management server.

This chapter contains the following topics:


- Configure Event Forwarding
- Configure Remote View Launching
- Test the Integration
- Troubleshoot Event Forwarding

# Configure Event Forwarding

To set up event forwarding from the NNM 6.x/7.x management station to an NNMi 8.10 management server, complete the following procedures in order:

- Step 1: Configure NNM 6.x/7.x to Forward Events to the NNMi 8.10 Management Server
- Step 2: (Optional) Use Node Level Filtering to Further Reduce Events
- Step 3: Add the NNM 6.x/7.x Management Station to the NNMi 8.10 Topology
- Step 4: (Optional) Save the Management Station Configuration
- Step 5: Verify NNM 6.x/7.x Incident Configuration in the NNMi 8.10 Console

## Step 1: Configure NNM 6.x/7.x to Forward Events to the NNMi 8.10 Management Server

On the NNM 6.x/7.x management station, configure each event that you want forwarded to the NNMi 8.10 management server. Most of these events will be under the OpenView Enterprise. Interesting events include:

- OV_Node_Down (OV_Node_Up, Ov_Node_Unknown, and so forth)
- OV_APA_NODE_DOWN (OV_APA_NODE_Intermittent, and so forth)
- OV_Station_Critical (OV_Station_Normal, and so forth)
- OV_Error (OV_Warning, OV_Inform) information about system health
- OV_Message (OV_Popup_Message, and so forth)

For the complete list of recommended NNM 6.x/7.x events to forward, see the events listed on the **Remote NNM 6.x/7.x Event Configuration** tab of the **Incident Configuration** form in the NNMi console.

### Recommended and Supported Procedure: Use the Event Configuration Window

If you do not have an XServer, see Alternative Procedure: Manually Edit trapd.conf on page 176.

To configure an NNM 6.x/7.x event to forward to the NNMi 8.10 management server, follow these steps:

1. At the command prompt, enter:

```
ovw
```

Alternatively, run `xnmtrap` from the command line, and then continue with step 3.

2. Click **Options > Event Configuration**.
3. In the **Event Configuration** window, select the **Openview** enterprise in the top pane, and then double-click an event name in the bottom pane.

To sort the events by name, click **View > Sort > Event Name**.

4. Specify the NNMi 8.10 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 175.

- *Windows*: On the **Forwarding** tab in the **Modify Events** window, enter the host name of the NNMi 8.10 management server in the **Destination** field.

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

- *UNIX*: In the **Destination** field at the bottom of the **Event Configurator** window, enter the host name of the NNMi 8.10 management server.

If you do not see the **Destination** field, select the **Forward Event** option in the center of the window.

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

5 Repeat step 3 and step 4 until you have configured all of the events that you want to forward to an NNMi 8.10 management server.

6 Click **File > Save**.

NNM 6.x/7.x saves the changes to the event configurations and automatically re-reads the new event configuration.

## Optional: Destination List File

If you want to forward several events to the same group of NNMi 8.10 management servers, you can create a file that lists the forward destinations.

The recommended location for the destination list file is:

- Windows: `%OV_CONF%\nnm8EventForwardDestinations.txt`
- UNIX: `$OV_CONF/nnm8EventForwardDestinations.txt`

The destination list file is a text file with the following format:

- Each line is either one node name or a comment line.
- The first character of a comment line is the # character.

For example:

```
# List of destination NNMi 8.10 Management Servers to receive events.
# This list should be small enough that it does not overwhelm the NNMi 8.10 operators.
# In general, the events should be node-related, so that Neighbor Views launched remotely
from the NNMi 8.10 management server are meaningful.
#
system1.domain.com
system2.comain.com
system3.domain.com
```

For more information, see the `trapd.conf` manpage.

After creating or changing the destination list file, run the following command to re-read it:

```
xnmevents -event
```

### Alternative Procedure: Manually Edit trapd.conf

If you do not have an XServer, you can manually edit the `FORWARD` field for each event in the following file:

- Windows: `%OV_CONF%\C\trapd.conf`
- UNIX: `$OV_CONF/C/trapd.conf`

Specify either a single NNMi 8.10 management server or a destination list file. For example:

```
EVENT OV_Message .1.3.6.1.4.1.11.2.17.1.0.58916872 "Application Alert Alarms" Normal
FORMAT $3
FORWARD NNM8Server.domain.com
```

The `FORWARD` field might also include a list of the remote managers. For example:

```
FORWARD %REMOTE_MANAGERS_LIST% /etc/opt/OV/share/conf/nnm8EventForwardDestinations.txt
```

After editing the `trapd.conf` file, run the following command to force NNM to re-read the event configuration:

```
xnmevents -event
```

## Step 2: (Optional) Use Node Level Filtering to Further Reduce Events

In NNM 6.x/7.x, you can configure a node list for certain events. When a node list is present, an event coming into the NNM 6.x/7.x management station matches an event configuration only if the event source is in the node list. Thus, an event will be forwarded to the NNMi 8.10 management server only if the event source is in the node list. A typical use case for a node list is to forward only specific events from important nodes to the NNMi 8.10 management server.

For information on creating a node list in NNM 6.x/7.x, see the information about the `sources_list` in the `ovtrapd.conf` manpage.

## Step 3: Add the NNM 6.x/7.x Management Station to the NNMi 8.10 Topology

Include the NNM 6.x/7.x management station in the NNMi 8.10 topology so that the NNMi 8.10 management server receives an incident if the NNM 6.x/7.x management station goes down.

If the NNM 6.x/7.x management station is not already in the NNMi 8.10 **Nodes** inventory view, add the management station to the discovery seeds, and then wait for it to be discovered.

For information on how to add a node to the discovery seeds, refer to *Discovering Your Network* in the NNMi help.

## Step 4: (Optional) Save the Management Station Configuration

To save the new configuration, run the following command:

```
nnmconfigexport.ovpl -u <user> -p <password> -c station
-f <filename>
```

You can later import the backup by running the following command:

```
nnmconfigimport.ovpl -u <user> -p <password> -f <filename>
```

For information on these commands, refer to their respective reference pages, or the UNIX manpages.

## Step 5: Verify NNM 6.x/7.x Incident Configuration in the NNMi 8.10 Console

Verify that the events you forwarded from NNM 6.x/7.x are configured (as incidents) in NNMi 8.10.

NNMi 8.10 provides many out-of-the-box incident configurations. To view these out-of-the-box configurations, follow these steps:

1. From the workspace navigation panel, select the **Configuration** workspace.
2. Open the **Incident Configuration** form.
3. Click the **Remote NNM 6.x/7.x Event Configuration** tab.

   This tab displays the out-of-the-box incident configurations. If you open an incident, you should see that the type is **Remotely Generated**.

If one or more of the events that you configured for forwarding from the NNM 6.x/7.x management station is not listed on the **Remote NNM 6.x/7.x Event Configuration** tab, add a new incident configuration for each missing event. For information about how to configure a new incident, refer to *Configuring Incidents* in the NNMi help.

The incident categories in NNM 6.x/7.x are different from those in NNMi 8.10. For information about the relationship between the NNM 6.x/7.x alarm categories and the NNMi 8.10 incident categories, see Mapping Categories.

### Mapping Categories

In NNM 6.x/7.x, the pre-configured alarm categories are as follows:

- Error Alarms
- Threshold Alarms
- Status Alarms
- Configuration Alarms
- Application Alert Alarms

In NNMi 8.10, the pre-configured incident categories are as follows:

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

HP suggests the following mapping of NNM 6.x/7.x alarm categories to NNMi 8.10 incident categories:

| NNM 6.x/7.x Alarm Category | NNMi 8.10 Incident Category |
|---|---|
| Error Alarms | Application Status |
| Threshold Alarms | Performance |
| Status Alarms | Status |
| Configuration Alarms | Configuration |
| Application Alert Alarms | Application Status |

# Configure Remote View Launching

To set up the NNMi 8.10 management server to display NNM 6.x/7.x views on the NNMi 8.10 management server, complete the following procedures in order:

- Step 1: Install Java Run-Time Environment (JRE)
- Step 2: Create an NNM 6.x/7.x Management Station Entity in NNMi 8.10
- Step 3: (Optional) Configure Additional NNM 6.x/7.x Views

## Step 1: Install Java Run-Time Environment (JRE)

Although NNMi 8.10 does not have any requirements for a Java Run-Time Environment (JRE), NNM 6.x/7.x views require version 1.4.2.xx of the JRE. To download and install the required JRE, go to:

**http://java.sun.com/j2se/1.4.2/download.html**

NNM 7.53 does support newer JRE versions. Refer to the NNM 7.53 release notes for more information.

## Step 2: Create an NNM 6.x/7.x Management Station Entity in NNMi 8.10

Configure the NNMi 8.10 management server to associate events received from the NNM 6.x/7.x management station to an entity in NNMi 8.10. This configuration enables the launching of NNM 6.x/7.x Dynamic Views from the NNMi 8.10 management server. For example, you can select a Node Down from My7xSystem that is displayed in NNMi 8.10, and then launch the URLs back to My7xSystem.

It is important to use the Primary Address that matches the address that is encoded in the event sent by the NNM 6.x/7.x management station. If you are not sure about this address, look at the **RemoteSenderAddress** in the custom incident attributes for an incident that was forwarded from the NNM 6.x/7.x management station.

To set up an NNM 6.x/7.x management station configuration, follow these steps:

1. In the NNMi 8.10 console, from the workspace navigation panel, select the **Configuration** workspace.
2. Open the **Management Stations** form.
3. Click **New**.
4. Enter the following information:
   - **Name**—An identifier for the NNM 6.x/7.x management station represented by this configuration.
   - **NNM Version**—The NNM version (6.x or 7.x) of the management station that you are configuring.
   - **Address**—An IP address for the NNM 6.x/7.x management station. This IP address much be reachable from the NNMi 8.10 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 174 and if a configured event has been generated on the NNM 6.x/7.x management station and forwarded to the NNMi 8.10 management server.

   - **ovas (OV Application Server) Port**—The port number of the OpenView Application Server (ovas) for the NNM 7.x management station that you are configuring. On NNM 7.x management stations, the port number is usually 7510.

The ovas port also applies to NNM 6.x with the Extended Topology add-on.

   - **Web Server Port**—The port number of the web server for the NNM 6.x/7.x management station that you are configuring:
     - For NNM 6.x management stations on a Windows operating system, this port number is usually 80.
     - For NNM 6.x management stations on a UNIX operating system, this port number is usually 3443.
     - For NNM 7.x management stations on all operating systems, this port number is usually 3443.
   - **Description**—A description of the NNM 6.x/7.x management station that you are configuring.
5. Click **Save and Close**.

## Step 3: (Optional) Configure Additional NNM 6.x/7.x Views

The following URLs are not added out-of-the-box. You can add any of these URLs to the NNM 6.x/7.x deployment.

### URLs That Do Not Require a Selection

- MIB Browser Example URL:

```
http://192.168.1.xxx:3443/OvCgi/OpenView5.exe?Action=Snmp&Host=speed2.cnd.hp.com
```

- Report Presenter Example URL:

```
http://192.168.1.xxx:3443/OvCgi/nnmRptPresenter.exe
```

- Topology Summary Example URL:

```
http://192.168.1.xxx:7510/topology/summary
```

- SNMP Data Presenter (MIB Form/Table contrib. graphs):

```
http://192.168.1.xxx:3443/OvCgi/snmpviewer.exe?Context=Performance&sel=10.97.245.242
```

- OV Launcher Example URL:

```
http://system.yourcompany.com:3443/OvCgi/ovlaunch.exe
```

- jovw Example URL:

  (Web-based ovw, requires an ovw session running; otherwise, you see the error message `"Cannot find an ovw on host ..."` with map named default using sessionID xxxx:x):

```
http://system.yourcompany.com:3443/OvCgi/jovw.exe
```

  This URL can take a context node and map name, with option such as: `jovw.exe?mapName=default&ObjectName=10.1.12.33`

- ovalarm Example URL:

```
http://system.yourcompany.com:3443/OvCgi/ovalarm.exe
```

- Form to request topology details (type in a node by Name, IP Address, Physical Address UUID, OvwId):

```
http://192.168.1.xxx:7510/topology/topoDetail
```

### URLs That Require a Selection

- Node Details using an ovwId:

```
http://192.168.1.xxx:7510/topology/
topoDetail?objectType=ovwId&objectValue=3&Show+Details=Show+Details
```

- Node Details using a UUID:

```
http://192.168.1.xxx:7510/topology/
topoDetail?objectType=uuid&objectValue=3dasfasdf&Show+Details=Show+Details
```

## Test the Integration

To verify that you have correctly set up the NNM 6.x/7.x integration with the NNMi 8.10 management server, complete one or both of the following procedures, as appropriate:

- Test 1: Verify Event Forwarding
- Test 2: Launch NNM 6.x/7.x Dynamic Views from NNMi 8.10

## Test 1: Verify Event Forwarding

To verify event forwarding from the NNM 6.x/7.x management station to the NNMi 8.10 management server, follow these steps:

1. On the NNM 6.x/7.x management station, create a situation that generates one of the forwarded events.

   The simplest approach is to run the `sendMsg.ovpl` command on the NNM 6.x/7.x management station. For information on how to run this command, see sendMsg.ovpl.

   Another approach would be to generate or simulate a network fault on the NNM 6.x/7.x system. See Generate Node/Interface Down/Up Events on page 181.

2. In the NNMi 8.10 console, from the workspace navigation panel, select the **Incidents** workspace.

3. Select **NNM 6.x/7.x Events**.

   The event that you generated from the NNM 6.x/7.x management station should be visible in this view.

Alternatively, you can run **`nnmdumpevents -t`** on the NNMi 8.10 management server to see the list of events that the NNMi 8.10 management station has received.

### Generate Node/Interface Down/Up Events

To generate a false interface down event, which can cause a node down event, run the following command:

**`ovtopofix -S Down <IPADDR>`**

Where ***<IPADDR>*** is the IP address of one of the interfaces that has been discovered on the NNM 6.x/7.x management station.

Run the ovtopofix command once for each IP interface on the node. To determine the IP addresses to use, run the following command:

**`ovtopodump > topology.txt`**

In the `topology` 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.

After generating the interface and node down events, run the following command once for each IP interface to generate Interface Up and Node Up events:

```
ovtopofix -S Up <IPADDR>
```

Make sure that the events you are testing (in this case, OV_IF_Up/OV_IF_Down, which are .1.3.6.1.4.1.11.2.17.1.0.58916866 and .1.3.6.1.4.1.11.2.17.1.0.58916867, respectively) are configured to be forwarded to the NNMi 8.10 management server. Also, make sure that you have formatted rules for these events; otherwise, they will appear as `unknown` in the incident views.

## sendMsg.ovpl

You can run the `sendMsg.ovpl` command to generate an OV_Message event.

For example:

- Windows:

```
%OV_CONTRIB%\NNM\sendMsg\sendMsg.ovpl "" "Test from `hostname` on `date`"
```

- UNIX:

```
$OV_CONTRIB/NNM/sendMsg/sendMsg.ovpl "" "Test from `hostname` on `date`"
```

Each time you run the `sendMsg.ovpl` command, NNM generates an OV_Message event containing the text that you included in the `sendMsg.ovpl` command line. For example:

```
1183160690 6 Fri Jun 29 17:44:50 2007 <none>    a Test from
speed2 on Fri Jun 29 17:44:50 MDT 2007;1 17.1.0.58916872 0
```

This event is visible in the **All Alarms** browser on the NNM 6.x/7.x management station.

Best practice

To facilitate identification of new alarms, delete all of the alarms in the **All Alarms** browser before running the `sendMsg.ovpl` command.

By default, the OV_Message incident is not configured in NNMi 8.10. In order to run this test you must have configured forwarding for the OV_Message event in NNM 6.x/7.x, and you must have configured the OV_Message incident in the NNMi 8.10 **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 traps that are received being forwarded also.

You can manually generate traps on the NNM 6.x/7.x management station with a command such as:

```
snmptrap -p 162 Hostname "" "" 6 1234 "" .1.3.6.1.3.1.1.5.3
octetstring "Test Trap"
```

The example generates a SNMP_Link_Down trap. Use the event object identifier for a trap that you configured to be forwarded.

For more information, see the *snmptrap* manpage. Hostname is the name of the NNM 6.x/7.x system.

### Test 2: Launch NNM 6.x/7.x Dynamic Views from NNMi 8.10

1. In the NNMi 8.10 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

2. Open each of the views from the **Actions** menu.

## Troubleshoot Event Forwarding

If you did not see the expected NNM 6.x/7.x events in the **NNM 6.x/7.x Events** view, follow these steps to troubleshoot the problem:

1. On the NNM 6.x/7.x management station, run the following command:

   **`ovdumpevents -t -l <n>`**

   Where ***<n>*** specifies the number of minutes to go back in the event history. For example, when the value for *n* is 1, the `ovdumpevents` command displays the events that have been generated on the NNM 6.x/7.x management station in the last *n* minutes.

2. If an expected event is not included in the `ovdumpevents` output, the event was not generated. Refer to the NNM 6.x/7.x documentation for information about troubleshooting this situation.

3. Repeat step 1 until all expected events are included in the `ovdumpevents` output on the NNM 6.x/7.x management station.

4. On the NNMi 8.10 management server, run the following command:

   **`nnmdumpevents -t -l <n>`**

   Where ***<n>*** specifies the number of minutes to go back in the event history. For example, when the value for *n* is 1, the `nnmdumpevents` command displays the events that have been generated on the NNMi 8.10 management server in the last *n* minutes.

5. For each expected event that is not included in the `nnmdumpevents` output, verify the configuration of that event in the **Event Configurator** window on the NNM 6.x/7.x management station.

   - Verify that the **Forward Event** option is selected.
   - Verify the names or IP addresses of the NNMi 8.10 management servers in the **Forwarded Event Destinations** list.

   For more information, see Step 1: Configure NNM 6.x/7.x to Forward Events to the NNMi 8.10 Management Server on page 174.

6 Repeat step 5 until all expected events are included in the `nnmdumpevents` output on the NNMi 8.10 management server.

7 In the NNMi 8.10 console, examine the **NNM 6.x/7.x Events** incident view. If the results are not as expected, verify the incident configuration from the **Remote NNM 6.x/7.x Event Configuration** tab of the **Incident Configuration** form.

# Integrations and Plug-ins

This section contains the following chapters:


- HP Operations Manager
- HP Universal Configuration Management Database
- AlarmPoint
- nGenius Performance Manager
- NNM iSPI for Performance

# HP Operations Manager

HP Operations Manager (HPOM) provides comprehensive event management; proactive performance monitoring; and automated alerting, reporting, and graphing for management operating systems, middleware, and application infrastructure. HPOM consolidates events from a wide range of sources into a single view.

For information about purchasing HPOM, contact your HP sales representative.

This chapter contains the following topics:


- HP NNMi–HPOM Integration
- Enabling the HP NNMi–HPOM Integration
- Using the HP NNMi–HPOM Integration
- Changing the NNMi–HPOM Integration Configuration
- Disabling the HP NNMi–HPOM Integration
- Troubleshooting the HP NNMi–HPOM Integration
- HP NNMi–HPOM Integration Configuration Form Reference
- HP NNMi–HPOM Integration Log File Reference


## HP NNMi–HPOM Integration

The NNMi–HPOM integration forwards NNMi incidents to the HPOM active messages browser. The integration synchronizes incidents between NNMi and HPOM. It also provides for accessing the NNMi console from within HPOM.

The NNMi–HPOM integration supports a “many-to-many” arrangement. Each NNMi management server can forward incidents to multiple HPOM management servers. Likewise, each HPOM management server can receive incidents from multiple NNMi management servers. The integration interprets the unique identifier of an incident to determine the source NNMi management server.

The NNMi–HPOM integration consists of the following components:

- **NNMi–HPOM Integration Module**

  The NNMi–HPOM integration module forwards incidents from NNMi to HPOM. It is installed and configured on the NNMi management server.

- **HP Operations Manager Incident Web Service**

  HPOM uses the HP Operations Manager Incident Web Service (IWS) to receive the incidents that are forwarded from NNMi.

- **HPOM applications for contextual access of the NNMi console**

  HPOM provides applications for accessing forms, views, and tools in the NNMi console. For example, you can open an NNMi incident directly from the HPOM active messages browser. The specific application determines the context in which the NNMi console opens. You need to configure the applications before you can use them.

## Value

The NNMi–HPOM integration provides event consolidation in the HPOM active messages browser for the network management, system management, and application management domains, so that HPOM users can detect and investigate potential network problems.

The primary features of the integration are as follows:

- Automatic incident forwarding from NNMi to HPOM.
  - Forwarded incidents appear in the HPOM active messages browser.
  - You can create filters that limit which incidents NNMi forwards.
- Synchronization of incident updates between NNMi and HPOM as described in the following table.

| Trigger | Result |
|---|---|
| In HPOM, the message is acknowledged. | In NNMi, the corresponding incident's lifecycle state is set to Closed. |
| In HPOM, the message is unacknowledged. | In NNMi, the corresponding incident's lifecycle state is set to Registered. |
| In NNMi, the incident's lifecycle state is set to Closed. | In HPOM, the corresponding message is acknowledged. |
| In NNMi, the incident's lifecycle state is changed from Closed to any other state. | In HPOM, the corresponding message is unacknowledged. |

- Access to the NNMi console from HPOM.
  - HPOM users can open the NNMi Incident form in the context of a selected message.
  - HPOM users can launch an NNMi view (for example, the Layer 2 Neighbor view) in the context of a selected message and node.

— HPOM users can launch an NNMi tool (for example, status poll) in the context of a selected message and node.

— When HPOM is consolidating NNMi incidents from multiple NNMi management servers, the integration interprets the unique identifier of each incident to access the correct NNMi management server.

## Supported Versions

The information in this chapter applies to the following product versions:

- HPOM for Windows version 8.10 or higher
- HPOM for UNIX version 8.30 or higher
- NNMi version 8.10 or higher

For information about how to configure the HP NNMi–HPOM for NNMi 8.0x (starting with NNMi 8.03), see the chapter *Integration with HP Operations Manager* in the *NNMi Deployment Guide* that corresponds to your version of NNMi.

NNMi and HPOM cannot be installed on the same computer. The two products must be installed on different computers in either of the following configurations:

- Different operating systems. For example, the NNMi management server is a Linux system, and the HPOM management server is a Windows system.
- The same operating system. For example, the NNMi management server is a Windows system, and the HPOM management server is a second Windows system.

For the most recent information about supported hardware platforms and operating systems, refer to the support matrices for both products.

## Documentation

This chapter describes how to configure NNMi to communicate with HPOM.

The HPOM documentation describes how to configure HPOM to communicate with NNMi. It also describes how to use the HP NNMi–HPOM integration.

- For HPOM for Windows, see the information for the HP NNMi Adapter in the HPOM online help.
- For HPOM for UNIX, see the *HP NNMi–HPOM Integration for HP Operations Manager User's Guide*.

# Enabling the HP NNMi–HPOM Integration

This section describes the procedure for enabling the NNMi–HPOM integration. For each NNMi management server and each HPOM management server that you want to include in the integration, complete the appropriate steps in the procedure for the version of HPOM that you are using.

## HPOM for Windows

1. On the NNMi management server, configure NNMi incident forwarding to HPOM:
   a. In the NNMi console, open the **HP NNMi–HPOM Integration Configuration** form (**Integration Module Configuration > HPOM**).
   b. Select the **Enable Integration** check box to activate the remaining fields in the form.
   c. Enter the information for connecting to the NNMi management server.

      For information about these fields, see NNMi Management Server Connection on page 198.
   d. Enter the information for connecting to the HPOM management server.

      For information about these fields, see HPOM Management Server Connection on page 199.
   e. If you want NNMi to forward incidents to multiple HPOM management servers, click **Add another OM server**, and then enter the information for the next HPOM management server in the HPOM fields.

      The information for the first server appears in the **Additional HPOM Servers** list.
   f. Click **Submit** at the bottom of the form.

      A new window displays a status message. If the message indicates a problem with connecting to the HPOM server, re-open the **HP NNMi–HPOM Integration Configuration** form (or press **ALT+LEFT ARROW** in the message window), and then adjust the values for connecting to the HPOM management server as suggested by the text of the error message.
2. On the NNMi management server, customize the integration:
   a. In the NNMi console, open the **HP NNMi–HPOM Integration Configuration** form (**Integration Module Configuration > HPOM**).
   b. Enter values for the following fields:
      - **Forward Only**
      - **Holding period (minutes)**
      - **Incident Filter**

      For information about these fields, see Integration Behavior on page 200.
   c. Click **Submit** at the bottom of the form.
3. In HPOM, configure the NNMi adapter for connecting to the NNMi management server as described in *Configure the NNMi Server Name and Port* of the HPOM for Windows online help.
4. In HPOM, create a managed node for each NNMi node that will be named as a source node in the NNMi incidents that are forwarded to this HPOM management server. Also create a managed node for each NNMi management server that will forward incidents to this HPOM management server.

   Alternatively, you can create one external node to catch all forwarded NNMi incidents.

For more information, see *Configuring NNMi Server Nodes* in the HPOM online help.

If you do not set up an HPOM managed node for an NNMi incident source node, the HPOM management server discards all incidents regarding that node.

5 In HPOM, add the custom message attributes for NNMi incidents to the active messages browser:

   a In the browser, right-click any column heading, and then click **Options**.

   b In the **Enter Custom Message Attributes** list, select an attribute, and then click **Add**.

      — The custom message attributes for NNMi incidents begin with the text `nnm`.

      — The most interesting attributes for NNMi incidents are as follows:

```
nnm.assignedTo
nnm.category
nnm.emittingNode.name
nnm.source.name
```

      — To change the order in which the custom message attributes appear in the messages browser, drag a column heading to the new location.

6 In HPOM, enable contextual launching of the NNMi views by associating the NNMi source nodes with the HP NNMi Web Tools group.

   For more information, see *Enabling HP NNMi Web Tools in the By Node Group* in the HPOM online help.

## HPOM for UNIX

1 Prepare the HPOM for UNIX management server:

   a On the HPOM for UNIX management server, install the HP Operations Manager Incident Web Service (IWS) as described in the *HP Operations Manager Incident Web Service Integration Guide*.

   b On the HPOM for UNIX management server, install the most recent HPOM consolidated patch, which is available at:

      **http://h20230.www2.hp.com/selfsolve/patches**

2 On the NNMi management server, configure NNMi incident forwarding to HPOM:

   a In the NNMi console, open the **HP NNMi–HPOM Integration Configuration** form (**Integration Module Configuration > HPOM**).

   b Select the **Enable Integration** check box to activate the rest of the fields in the form.

   c Enter the information for connecting to the NNMi management server.

      For information about these fields, see NNMi Management Server Connection on page 198.

   d Enter the information for connecting to the HPOM management server.

      For information about these fields, see HPOM Management Server Connection on page 199.

e If you want NNMi to forward incidents to multiple HPOM management servers, click **Add another OM server**, and then enter the information for the next HPOM management server in the HPOM fields.

The information for the first server appears in the **Additional HPOM Servers** list.

f Click **Submit** at the bottom of the form.

A new window displays a status message. If the message indicates a problem with connecting to the HPOM server, re-open the **HP NNMi–HPOM Integration Configuration** form (or press **ALT+LEFT ARROW** in the message window), and then adjust the values for connecting to the HPOM management server as suggested by the text of the error message.

3 On the NNMi management server, customize the integration:

a In the NNMi console, open the **HP NNMi–HPOM Integration Configuration** form (**Integration Module Configuration > HPOM**).

b Enter values for the following fields:

— **Forward Only**

— **Holding period (minutes)**

— **Incident Filter**

For information about these fields, see Integration Behavior on page 200.

c Click **Submit** at the bottom of the form.

4 In HPOM, create a managed node for each NNMi node that will be named as a source node in the NNMi incidents that are forwarded to this HPOM management server. Also create a managed node for each NNMi management server that will forward incidents to this HPOM management server.

Alternatively, you can create one external node to catch all forwarded NNMi incidents.

For more information, see the *HP Operations Manager for UNIX Administrator's Reference*.

If you do not set up an HPOM managed node for an NNMi source node, the HPOM management server discards all incidents regarding that node.

5 In HPOM, add the custom message attributes for NNMi incidents to the active messages browser:

a In the browser, right-click any column heading, and then click **Customize Message Browser Columns**.

b On the **Custom** tab, select from the **Available Custom Message Attributes**, and then click **OK**.

— The custom message attributes for NNMi incidents begin with the text `nnm`.

— The most interesting attributes for NNMi incidents are as follows:

```
nnm.assignedTo
nnm.category
nnm.emittingNode.name
nnm.source.name
```

— To change the order in which the custom message attributes appear in the messages browser, drag a column heading to the new location.

6 On the HPOM management server, prepare the HPOM applications for accessing the NNMi console.

a *Required*. Install the basic set of NNMi applications.

b *Optional*. Install additional NNMi applications.

For more information, see the chapter on installing and configuring the NNMi–HPOM integration in the *HP NNMi–HPOM Integration for HP Operations Manager User's Guide*.

# Using the HP NNMi–HPOM Integration

Figure 11 shows an address not responding incident in the NNMi console. The information in the **Source Object** and **Message** columns together describe the situation.

**Figure 11 Address Not Responding Incident in NNMi Console**

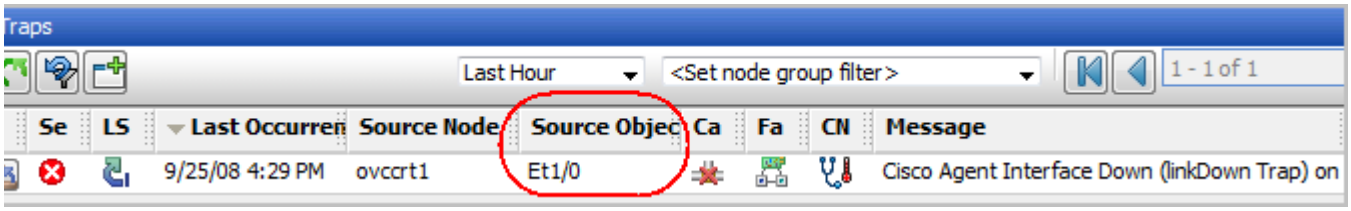



Figure 12 shows the NNMi incident as received by HPOM for Windows. Figure 13 shows the NNMi incident as received by HPOM for UNIX. The **nnm.source.name** and **Text** columns are equivalent to the **Source Object** and **Message** columns in the NNMi console.

You must enable the display of the **nnm.source.name** custom message attribute column as described in step 5 on page 191 (for HPOM for Windows) and in step 5 on page 192 (for HPOM for UNIX).

**Figure 12 Forwarded Incident in HPOM for Windows**

**Figure 13 Forwarded Incident in HPOM for UNIX**

For detailed information about using the NNMi–HPOM integration, see the HPOM documentation.

- For HPOM for Windows, see *NNMi Web Tools* in the HPOM online help.
- For HPOM for UNIX, see the chapter on using the NNMi applications in the *HP NNMi–HPOM Integration for HP Operations Manager User's Guide*.

➤ The NNMi help does not describe the new functionality that this integration adds to the NNMi console. Provide each NNMi operator with the integration information that is pertinent to that operator's job responsibilities.

➤ In the HPOM messages browser, the details for a forwarded NNMi incident are available as custom message attributes.

# Changing the NNMi–HPOM Integration Configuration

1. In the NNMi console, open the **HP NNMi–HPOM Integration Configuration** form (**Integration Module Configuration > HPOM**).
2. Modify the values as appropriate.
   - If you know the syntax of the entries in the Incident Filter and Additional HPOM Servers lists, you can modify the entries directly.
   - If you do not know the syntax for a list item, delete that entry and then re-enter it.

   For information about the fields on this form, see HP NNMi–HPOM Integration Configuration Form Reference on page 198.
3. Verify that the **Enable Integration** check box at the top of the form is selected, and then click **Submit** at the bottom of the form.

➤ The changes take effect immediately. You do not need to restart `ovjboss`.

# Disabling the HP NNMi–HPOM Integration

## For All HPOM Management Servers

To discontinue the forwarding of NNMi incidents to all HPOM management servers, follow these steps:

1. In the NNMi console, open the **HP NNMi–HPOM Integration Configuration** form (**Integration Module Configuration > HPOM**).
2. Clear the **Enable Integration** check box at the top of the form, and then click **Submit** at the bottom of the form.

➤ The changes take effect immediately. You do not need to restart `ovjboss`.

If necessary, repeat this process for all NNMi management servers.

### For One HPOM Management Server

To discontinue the forwarding of NNMi incidents to only one of the HPOM management servers, follow these steps:

1. In the NNMi console, open the **HP NNMi–HPOM Integration Configuration** form (**Integration Module Configuration > HPOM**).
2. In the **Additional HPOM Servers** list, edit the text to delete the entry (or entries) for the HPOM management server to disconnect from the integration.

Clicking **Clear** removes all HPOM servers from the list.

3. Click **Submit** at the bottom of the form.

The changes take effect immediately. You do not need to restart `ovjboss`.

## Troubleshooting the HP NNMi–HPOM Integration

### HPOM Does Not Receive Any Forwarded Incidents

If the integration has worked successfully in the past, it is possible that some aspect of the configuration, for example, the NNMi or HPOM user password, has changed recently. You may want to update the integration configuration as described in Changing the NNMi–HPOM Integration Configuration on page 194, before walking through this entire procedure.

1. In the NNMi console, open the **HP NNMi–HPOM Integration Configuration** form (**Integration Module Configuration > HPOM**).
2. To check the status of the integration, in the **HP NNMi–HPOM Integration Configuration** form, click **Submit** at the bottom of the form (without making any configuration changes).

   A new window displays a status message.

   - If the message indicates a problem with connecting to the HPOM server, NNMi and HPOM are not able to communicate. Continue with step 3 of this procedure.
   - If the message indicates success, the problem is most likely that HPOM is not configured to accept incidents from the devices that NNMi manages. HPOM ignores any forwarded incident from an NNMi source node that is not configured as a managed node in HPOM, as described in step 4 on page 190 (for HPOM for Windows) and in step 4 on page 192 (for HPOM for UNIX).
3. Verify that the connection to the HPOM management server is configured correctly:

   a. In a web browser, enter the following URL:

   **https://*<omserver>*:*<port>*/opr-webservice//Incident.svc?wsdl**

   Where `<omserver>` and `<port>` are the values of **HPOM Host** and **HPOM Port**, respectively, on the **HP NNMi–HPOM Integration Configuration** form.

b When prompted, enter the credentials for the **HPOM User** from the **HP NNMi–HPOM Integration Configuration** form.

The resulting web page is an XML file that describes the IWS.

— If the XML file appears, the connection to the HPOM management server is configured correctly. Continue with step 4.

— If you see an error message, the connection to the HPOM management server is not configured correctly. Contact the HPOM administrator to verify the information that you are using to connect to the HPOM web services. Continue to troubleshoot the connection to HPOM until you see the XML file.

4 Verify that the connection NNMi is configured correctly:

If you used the information described in this step to connect to the NNMi console in step 1 of this procedure, you do not need to reconnect to the NNMi console. Continue with step 5.

a In a web browser, enter the following URL:

**http://*<NNMiserver>*:*<port>*/nnm/**

Where *<NNMiserver>* and *<port>* are the values of **NNMi Host** and **NNMi Port**, respectively, on the **HP NNMi–HPOM Integration Configuration** form.

Use the fully-qualified domain name of the NNMi management server as returned by the following command:

```
nnmofficialfqdn.ovpl -t
```

b When prompted, enter the credentials for an NNMi user with the Administrator role.

You should see the NNMi console. If the NNMi console does not appear, contact the NNMi administrator to verify the information that you are using to connect to NNMi. Continue to troubleshoot the connection to NNMi until the NNMi console appears.

You cannot sign into the NNMi console as a user with the Web Service Client role.

c Verify the values of the **NNMi User** and **NNMi Password**.

— If the **NNMi User** listed on the **HP NNMi–HPOM Integration Configuration** form has the Administrator role and you were able to connect to the NNMi console with this user name, then re-enter the corresponding password on the **HP NNMi–HPOM Integration Configuration** form.

— If the **NNMi User** listed on the **HP NNMi–HPOM Integration Configuration** form has the Web Service Client role, contact the NNMi administrator to verify the values of **NNMi User** and **NNMi Password**.

Passwords are hidden in the NNMi console. If you are not sure what password to specify for an NNMi user name, ask the NNMi administrator to reset the password.

5 Update the **HP NNMi–HPOM Integration Configuration** form with the values that you used for successful connections in step 3 and step 4 of this procedure.

For more information, see HP NNMi–HPOM Integration Configuration Form Reference on page 198.

6 Click **Submit** at the bottom of the form.

7 Test the configuration by generating an incident on the NNMi management server and determine whether it reaches the HPOM management server.

## HPOM Does Not Receive Some Forwarded Incidents

Verify the HPOM nodes and the incident filter.

The HPOM management server must be configured to accept incidents from the devices that NNMi manages. HPOM ignores any forwarded incident from an NNMi source node that is not configured as a managed node in HPOM, as described in step 4 on page 190 (for HPOM for Windows) and in step 4 on page 192 (for HPOM for UNIX).

If the NNMi source node is configured as a managed node in HPOM, verify the incident filter configuration on the **HP NNMi–HPOM Integration Configuration** form. Then test the filter by generating an incident on the NNMi management server and determine whether it reaches the HPOM management server.

## NNMi Incident Information Is Not Available in the HPOM Messages Browser

The important information from NNMi incidents is passed to HPOM as custom message attributes. Add one or more custom messages attributes for NNMi incidents as described in step 5 on page 191 (for HPOM for Windows) and in step 5 on page 192 (for HPOM for UNIX).

## NNMi and HPOM Are Not Synchronized

If either of the management servers becomes unreachable, the incidents in the NNMi incident views and the HPOM active messages browser might become mismatched. The NNMi–HPOM integration can re-synchronize the incidents as described here.

- If an HPOM management server becomes unavailable to the NNMi–HPOM integration module, the integration module periodically checks for the availability of that HPOM management server and resumes incident forwarding when a connection can be re-established. When the connection to the HPOM management server is available, the integration module forwards any incidents that might have been missed while the HPOM management server was down.
- If the NNMi management server is unavailable when an HPOM user acknowledges or unacknowledges a forwarded incident, NNMi does not receive the change of state. NNMi and HPOM might show different states for this incident.

## The Integration Does Not Work Through a Firewall

Ensure that the NNMi management server is able to directly address the HPOM IWS by host and port.

# HP NNMi–HPOM Integration Configuration Form Reference

The **HP NNMi–HPOM Integration Configuration** form contains the parameters for communications between NNMi and HPOM. This form is available from the **Integration Module Configuration** workspace.

Only NNMi users with the Administrator role can access the **HP NNMi–HPOM Integration Configuration** form.

The **HP NNMi–HPOM Integration Configuration** form collects information for the following general areas:

- NNMi Management Server Connection
- HPOM Management Server Connection
- Integration Behavior
- Incident Filters

To apply changes to the integration configuration, update the values on the **HP NNMi–HPOM Integration Configuration** form, and then click **Submit**.

## NNMi Management Server Connection

Table 16 lists the parameters for connecting to the NNMi management server. This is the same information that you use to open the NNMi console. You can determine many of these values by examining the URL that invokes an NNMi console session. Coordinate with the NNMi administrator to determine the appropriate values for this section of the configuration form.

The default NNMi configuration uses http for connecting to the NNMi console. For information about configuring this connection to use https, see Enabling https for NNMi on page 73.

**Table 16 NNMi Management Server Connection Information**

| Field | Description |
|---|---|
| NNMi SSL Enabled | The connection protocol specification.<br>• If the NNMi console is configured to use https, select the **NNMi SSL Enabled** check box.<br>• If the NNMi console is configured to use http, clear the **NNMi SSL Enabled** check box. This is the default configuration. |
| NNMi Host | The fully-qualified domain name of the NNMi management server. This field is pre-filled with host name that was used to access the NNMi console. Verify that this value is the name that is returned by the **`nnmofficialfqdn.ovpl -t`** command run on the NNMi management server. |

**Table 16 NNMi Management Server Connection Information (cont'd)**

| Field | Description |
|---|---|
| NNMi Port | The port for connecting to the NNMi console. This field is pre-filled with the port that the jboss application server uses for communicating with the NNMi console, as specified in the following file:<br>• *Windows*: `%NnmDataDir%\shared\nnm\conf\nnm.ports.properties`<br>• *UNIX*: `$NnmDataDir/shared/nnm/conf/nnm.ports.properties`<br>For non-SSL connections, use the value of `jboss.http.port`, which is `80` or `8004` by default (depending on the presence of another web server when NNMi was installed).<br>For SSL connections, use the value of `jboss.https.port`, which is `443` by default. |
| NNMi User | The user name for connecting to the NNMi console. This user must have the NNMi Administrator or Web Service Client role. |
| NNMi Password | The password for the specified NNMi user. |

## HPOM Management Server Connection

Table 17 lists the parameters for connecting to the web services on the HPOM management server. Coordinate with the HPOM administrator to determine the appropriate values for this section of the configuration.

**Table 17 HPOM Management Server Connection Information**

| HPOM Server Parameter | Description |
|---|---|
| HPOM SSL Enabled | The connection protocol specification.<br>• If HPOM is configured to use https, select the **HPOM SSL Enabled** check box. This is the default configuration.<br>• If HPOM is configured to use http, clear the **HPOM SSL Enabled** check box. |
| HPOM Host | The fully-qualified domain name of the HPOM management server. |
| HPOM Port | The port for connecting to the HPOM web services. To determine which port number to specify, do the following on the HPOM management server:<br>• *Windows*: Examine the port settings in the IIS Manager, which is available from the **Start** menu, for example, **Start > Administrative Tools > Internet Information Services (IIS) Manager**.<br>• *UNIX*: Run the following command: **`ovtomcatbctl -getconf`**<br>This field is pre-filled with the value `443`, which is the default port for SSL connections to HPOM for Windows. For SSL connections to HPOM for UNIX, the default port is 8443. |
| HPOM User | A valid HPOM user account name with the HPOM Administrator role.<br>*Windows only*: On the Windows operating system, HPOM works through Microsoft Internet Information Services (IIS) to authenticate user credentials. Specify a Windows user in the format *`<Windows_domain>\<username>`*. |
| HPOM Password | The password for the specified HPOM user. |

## Integration Behavior

Table 18 lists the parameters that describe the integration behavior. Coordinate with the NNMi administrator to determine the appropriate values for this section of the configuration.

**Table 18 Integration Behavior Information**

| Field | Description |
|---|---|
| Forward Only | The behavior specification for the NNMi–HPOM integration module. By default, the integration module forwards incidents to and receives incident acknowledgements from the HPOM management servers identified on the **HP NNMi–HPOM Integration Configuration** form. You can disable the receipt of incident acknowledgements.<br>• For one-way communication (forward incidents to HPOM but ignore incident acknowledgements from HPOM), select the **Forward Only** check box.<br>• For two-way communication, leave the **Forward Only** check box cleared. This is the default behavior. |
| Holding period (minutes) | The number of minutes to wait before forwarding the configured incidents to HPOM. If an incident is closed during this time (for example, an SNMPLinkUp incident cancels an SNMPLinkDown incident), HPOM never receives that incident. If you want NNMi to forward incidents immediately, enter the value `0`.<br>The default value is 5 minutes. |
| Incident Filter | A filter based on NNMi incident attributes that limits incident forwarding. The default filter (`nature=ROOTCAUSE origin=MANAGEMENTSOFTWARE`) specifies all root cause incidents that are generated by NNMi. You can modify the filter to change which incidents are forwarded to HPOM. For more information, see Incident Filters. |

## Incident Filters

The incident filter is the combination of all entries in the **Incident Filter** list. Filter entries with the same attribute value expand the filter (logical OR). Filter entries with different attribute values restrict the filter (logical AND). All filter entries work together; you cannot create a filter of the format `(a AND b) OR c`. Example filter entries follow the procedure.

To create the incident filter, follow these steps:

1. In the NNMi console, open the **HP NNMi–HPOM Integration Configuration** form (**Integration Module Configuration > HPOM**).
2. To delete a filter entry, in the **Incident Filter** list, edit the text to delete the entry (or entries).

Clicking **Clear** removes all filter entries from the list.

3. To add an incident filter entry:
   a. Select an attribute from the **name** list. For the supported attributes, see the table in step c.

b Select the comparison operation to perform. Supported operators are:

— =

— !=

— <

— <=

— >

— >=

c Enter a comparison value. The following table lists the supported attributes and the acceptable values for each attribute.

| Attribute | Possible Values |
|---|---|
| name | Examine the incident configuration in the NNMi console to determine the available incident names. |
| nature | • ROOTCAUSE<br>• SECONDARYROOTCAUSE<br>• SYMPTOM<br>• SERVICEIMPACT<br>• STREAMCORRELATION<br>• NONE |
| origin | • MANAGEMENTSOFTWARE<br>• MANUALLYCREATED<br>• SYMPTOM<br>• REMOTELYGENERATED<br>• SNMPTRAP<br>• SYSLOG<br>• OTHER |
| family | • com.hp.nms.incident.family.Address<br>• com.hp.nms.incident.family.Interface<br>• com.hp.nms.incident.family.Node<br>• com.hp.nms.incident.family.OSPF<br>• com.hp.nms.incident.family.HSRP<br>• com.hp.nms.incident.family.AggregatePort<br>• com.hp.nms.incident.family.Board<br>• com.hp.nms.incident.family.Connection<br>• com.hp.nms.incident.family.Correlation |

| Attribute | Possible Values |
|---|---|
| category | • com.hp.nms.incident.category.Fault<br>• com.hp.nms.incident.category.Status<br>• com.hp.nms.incident.category.Config<br>• com.hp.nms.incident.category.Accounting<br>• com.hp.nms.incident.category.Performance<br>• com.hp.nms.incident.category.Security<br>• com.hp.nms.incident.category.Alert |
| severity | • NORMAL<br>• WARNING<br>• MINOR<br>• MAJOR<br>• CRITICAL |

4 Repeat step 3 until all filter entries are defined.

5 Click **Submit** at the bottom of the form.

## Example Incident Filters

### Forward NodeDown Incidents from NNMi to HPOM

```
name=NodeDown
```

### Forward NodeDown and InterfaceDown Incidents from NNMi to HPOM

```
name=NodeDown
name=InterfaceDown
```

### Forward CiscoLinkDown Incidents from NNMi to HPOM

```
name=CiscoLinkDown
```

### Forward NNMi Management Events with Severity of MAJOR or MINOR

```
origin=Management Software
severity=MAJOR
severity=MINOR
```

### Forward NNMi Incidents with Severity of at least MINOR and nature of ROOTCAUSE or SERVICEIMPACT

```
severity>=MINOR
nature=ROOTCAUSE
nature=SERVICEIMPACT
```

## Incident Filter Limitations

Because all filter entries combine to create one incident filter for the NNMi management server, the following limitations apply:

- The stated severity applies to all incidents. For example, to forward NodeDown incidents with a severity of MINOR or higher and InterfaceDown incidents with a severity of MAJOR, set the filter severity to `>=MINOR` and use HPOM logic to filter out the unwanted InterfaceDown messages.
- The incident filter does not provide a mechanism for limiting incident forwarding to specific source nodes. The HPOM managed node (or external node) configuration limits the forwarded incidents that HPOM accepts.

# HP NNMi–HPOM Integration Log File Reference

## Log Files

The following log files contain messages for the NNMi–HPOM integration:

- NNMi management server:
  - *Windows*:

    ```
    %NnmDataDir%\log\nnm\nnm-om.#.#.log
    %NnmDataDir%\log\nnm\jbossserver.#.#.log
    ```

  - *UNIX*:

    ```
    $NnmDataDir/log/nnm/nnm-om.#.#.log
    $NnmDataDir/log/nnm/jbossserver.#.#.log
    ```

In addition to information about the NNMi management server, the `nnm-om.#.#.log` file contains messages regarding errors that occurred on the HPOM management server.

- HPOM for Windows management server:

  ```
  %ovdatadir%\shared\server\log\om\incident-ws.trace.txt
  ```

- HPOM for UNIX management server:

  ```
  \var\opt\OV\log\om\incident_ws.0.en
  ```

For example message streams in the `nnm-om.#.#.log` file, see Example Log File Messages on page 205.

## Enabling More Detailed Messages in the nnm-om Log File

By default, the `nnm-om.#.#.log` file is set to capture messages through the FINE logging level. To enable the capture of all messages, run the following command:

- *Windows*:

  ```
  %NnmInstallDir%\support\nnmsetlogginglevel.ovpl
  com.hp.ov.nnm.om FINEST
  ```

- *UNIX*:

  ```
  $NnmInstallDir/support/nnmsetlogginglevel.ovpl
  com.hp.ov.nnm.om FINEST
  ```

## Enabling Integration Communication Messages in the jboss-log4j Log File

By default, the `jbossserver.#.#.log` file does not capture messages from the HP NNMi–HPOM integration module. To enable the logging of all communication between NNMi and HPOM, follow these steps:

1. Back up the following file:
   - *Windows*:

     `%NnmInstallDir%\nonOv\jboss\nms\server\nms\conf\jboss-log4j.xml`
   - *UNIX*:

     `$NnmInstallDir/nonOv/jboss/nms/server/nms/conf/jboss-log4j.xml`
2. Open the `jboss-log4j.xml` file in a text editor.
3. Search for the string `nnm-om`, and then comment out the lines that follow this string. For example:

```
<!-- Next 3 added due to nnm-om module -->
    <category name="org.apache.axis2">
        <priority value="WARN"/>
    </category>
    <category name="org.apache.commons.httpclient">
        <priority value="WARN"/>
    </category>
    <category name="httpclient.wire">
        <priority value="INFO"/>
    </category>
```

4. Restart `ovjboss`:

```
ovstop ovjboss
ovstart ovjboss
```

## Example Log File Messages

The examples in this section are extracted from the `nnm-om.#.#.log` file. The following examples are included:

- Good Start up Trace
- NNMi Incident Successfully Forwarded to HPOM for UNIX
- NNMi Incident Successfully Closed in HPOM for UNIX
- Incorrect HPOM Login Configuration
- Incorrect NNMi Login Configuration
- Acknowledgement Error from HPOM for Windows
- Acknowledgement Error from HPOM for UNIX
- Node Not Found Error

## Good Start up Trace

```
Sep 2, 2008 10:07:00.921 AM [ThreadID:1] ALL: com.hp.ov.nms.admin.log.NnmOmFileHandler
init: Logging Handler initialized to log to file pattern: "C:/Documents and Settings/All
Users/Application Data/HP/HP BTO Software\log\nnm\nnm-om.%g.%u.log" up to a limit of 5
files of 1 MB each. Initially handling loggers:"com.hp.ov.nnm.om".

Sep 2, 2008 10:19:05.234 AM [ThreadID:61] CONFIG: com.hp.ov.nnm.om.im.NnmOmModule init:
Initialized nnm: http://nnmserver.domain.com:80 with filter(/sys:onNotification/
arg0[(name='NodeDown' or name='InterfaceDown' or name='SNMPLinkDown')])

Sep 2, 2008 10:19:05.265 AM [ThreadID:61] CONFIG: com.hp.ov.nnm.om.im.NnmOmModule init:
Initialized om: https://omserver.domain.com:8444

Sep 2, 2008 10:19:05.281 AM [ThreadID:61] FINE: com.hp.ov.nnm.om.im.NnmOmModule$OmServer
setSubscription: OM unsubscribing from: omserver.domain.com

Sep 2, 2008 10:19:05.281 AM [ThreadID:61] CONFIG: com.hp.ov.nnm.om.im.NnmOmModule disable:
NNM-OM module disabled

Sep 2, 2008 10:19:31.187 AM [ThreadID:61] FINE: com.hp.ov.nnm.om.im.NnmOmModule
subscribeNnm: NNM Subscription details: SubscriptionManager: http://NNMSERVER:80/
nms-sdk-notify/manage
SubscriptionID: urn:jbwse:58ea7aa4-ec0-4ec2-9949-895f8ca34f46
Expires: Tue Sep 02 10:29:16 BST 2008
Notify To: EPR {address=http://nnmserver.domain.com:80/NnmOmEventSinkBeanService/
NnmOmEventSinkBean, refParam=null}

Sep 2, 2008 10:19:31.218 AM [ThreadID:61] FINE: com.hp.ov.nnm.om.im.NnmOmModule$OmServer
setSubscription: OM unsubscribing from: omserver.domain.com

Sep 2, 2008 10:19:46.093 AM [ThreadID:61] FINE: com.hp.ov.nnm.om.im.OmBean
subscribeIncidents: Listening to omserver.domain.com for incidents from:
nnmserver.domain.com

Sep 2, 2008 10:19:46.125 AM [ThreadID:61] FINE: com.hp.ov.nnm.om.im.NnmOmModule
subscribeOm: OM Subscription details: OM Subscription Keys:
   omserver.domain.com: f5433fc2-b55d-4334-b234-2861d6a65605

Sep 2, 2008 10:19:46.125 AM [ThreadID:61] CONFIG: com.hp.ov.nnm.om.im.NnmOmModule enable:
NNM-OM module enabled

Sep 2, 2008 10:21:46.218 AM [ThreadID:63] FINE: com.hp.ov.nnm.om.im.NnmOmModule
doSweepForMissedIncidents: Sweeping for missed incidents for: omserver.domain.com

Sep 2, 2008 10:21:46.234 AM [ThreadID:63] FINE: com.hp.ov.nnm.om.im.NnmOmModule
doSweepQuery: Sweep query: select inc.uuid from Incident inc where inc.lifecycleState <>
4294967299 and (name='NodeDown' or name='InterfaceDown' or name='SNMPLinkDown') and
inc.id not in (select distinct i.id from Incident i, CustomIncidentAttribute cia where
cia.incident = i.id and cia.name like 'nnmOmEPR.omserver.domain.com')

Sep 2, 2008 10:22:00.890 AM [ThreadID:63] FINE: com.hp.ov.nnm.om.im.NnmOmModule
doSweepForMissedIncidents: Forwarding 1 incidents to: omserver.domain.com

Sep 2, 2008 10:22:44.453 AM [ThreadID:63] FINE: com.hp.ov.nnm.om.im.NnmOmModule
forwardNnmIncidentToOmServers: Forwarding to OM(omserver.domain.com), incident
08de1d45-5c57-4b61-a0d0-b9da119f066d

Sep 2, 2008 10:22:45.671 AM [ThreadID:63] FINE: com.hp.ov.nnm.om.im.NnmOmModule
forwardNnmIncidentToOmServers: Forwarded incident(08de1d45-5c57-4b61-a0d0-b9da119f066d)
response: https://omserver.domain.com:8444/opr-webservice/Incident.svc/
230e5b72-78cc-71dd-1875-1019f8490000

Sep 2, 2008 10:22:48.484 AM [ThreadID:67] FINE: com.hp.ov.nnm.om.im.NnmOmModule
onOmNotification: onNotification(omserver.domain.com):
08de1d45-5c57-4b61-a0d0-b9da119f066d/open
```

## NNMi Incident Successfully Forwarded to HPOM for UNIX

```
Sep 3, 2008 12:53:23.290 PM [ThreadID:61] FINE: com.hp.ov.nnm.om.im.NnmOmModule
onNewIncident: Forwarding incident(77f7bab0-7963-71dd-0142-ac1107ce0000) to:
omserver.domain.com:443

Sep 3, 2008 12:53:23.431 PM [ThreadID:61] FINE: com.hp.ov.nnm.om.im.NnmOmModule
onNewIncident: Forwarded incident(77f7bab0-7963-71dd-0142-ac1107ce0000) response: https://
omserver.domain.com/opr-webservice/Incident.svc/833cd1a2-630b-4ba1-956a-b5450bd15dce
```

## NNMi Incident Successfully Closed in HPOM for UNIX

```
Sep 2, 2008 12:46:37.234 PM [ThreadID:91] FINE: com.hp.ov.nnm.om.im.NnmOmModule
onOmNotification: onNotification(omserver.domain.com):
9a687f90-78e0-71dd-0713-1019f86f0000/closed

Sep 2, 2008 12:46:40.625 PM [ThreadID:91] FINE: com.hp.ov.nnm.om.im.NnmOmModule
synchronizeIncidentLifecycleState: Synchronizing NNM incident lifecycleState:
9a687f90-78e0-71dd-0713-1019f86f0000/com.hp.nms.incident.lifecycle.Closed
```

## Incorrect HPOM Login Configuration

```
Sep 2, 2008 10:54:46.500 AM [ThreadID:1] ALL: com.hp.ov.nms.admin.log.NnmOmFileHandler
init: Logging Handler initialized to log to file pattern: "C:/Documents and Settings/All
Users/Application Data/HP/HP BTO Software\log\nnm\nnm-om.%g.%u.log" up to a limit of 5
files of 1 MB each. Initially handling loggers:"com.hp.ov.nnm.om".

Sep 2, 2008 11:02:44.093 AM [ThreadID:61] CONFIG: com.hp.ov.nnm.om.im.NnmOmModule init:
Initialized nnm: http://nnmserver.domain.com:80 with filter(/sys:onNotification/
arg0[(name='NodeDown' or name='InterfaceDown' or name='SNMPLinkDown')])

Sep 2, 2008 11:02:44.109 AM [ThreadID:61] CONFIG: com.hp.ov.nnm.om.im.NnmOmModule init:
Initialized om: https://omserver.domain.com:8444

Sep 2, 2008 11:02:44.140 AM [ThreadID:61] FINE: com.hp.ov.nnm.om.im.NnmOmModule$OmServer
setSubscription: OM unsubscribing from: omserver.domain.com

Sep 2, 2008 11:02:44.140 AM [ThreadID:61] CONFIG: com.hp.ov.nnm.om.im.NnmOmModule disable:
NNM-OM module disabled

Sep 2, 2008 11:03:07.953 AM [ThreadID:61] FINE: com.hp.ov.nnm.om.im.NnmOmModule
subscribeNnm: NNM Subscription details: SubscriptionManager: http://NNMSERVER:80/
nms-sdk-notify/manage
SubscriptionID: urn:jbwse:d216f05e-caad-4b59-951b-89c57cb1d7b7
Expires: Tue Sep 02 11:12:57 BST 2008
Notify To: EPR {address=http://nnmserver.domain.com:80/NnmOmEventSinkBeanService/
NnmOmEventSinkBean, refParam=null}

Sep 2, 2008 11:03:07.968 AM [ThreadID:61] FINE: com.hp.ov.nnm.om.im.NnmOmModule$OmServer
setSubscription: OM unsubscribing from: omserver.domain.com

Sep 2, 2008 11:03:21.796 AM [ThreadID:61] SEVERE: com.hp.ov.nnm.om.im.NnmOmModule enable:
Error subscribing:  : org.apache.axis2.AxisFault: Transport error: 401 Error: No
AuthenticationProvider found for {0}
at org.apache.axis2.transport.http.HTTPSender.handleResponse(HTTPSender.java:298)
at org.apache.axis2.transport.http.HTTPSender.sendViaPost(HTTPSender.java:192)
at org.apache.axis2.transport.http.HTTPSender.send(HTTPSender.java:77)
at org.apache.axis2.transport.http.CommonsHTTPTransportSender.writeMessageWithCommons(
  CommonsHTTPTransportSender.java:327)
at org.apache.axis2.transport.http.CommonsHTTPTransportSender.invoke(
  CommonsHTTPTransportSender.java:206)
at org.apache.axis2.engine.AxisEngine.send(AxisEngine.java:396)
```

```
. . .

Sep 2, 2008 11:03:21.796 AM [ThreadID:61] CONFIG: com.hp.ov.nnm.om.im.NnmOmModule enable:
NNM-OM module enabled

Sep 2, 2008 11:04:26.406 AM [ThreadID:64] FINE: com.hp.ov.nnm.om.im.NnmOmModule$OmServer
setSubscription: OM unsubscribing from: omserver.domain.com

Sep 2, 2008 11:04:36.328 AM [ThreadID:64] SEVERE: com.hp.ov.nnm.om.im.NnmOmModule
checkOmSubscriptions: Failure subscribing to OM (omserver.domain.com):  :
org.apache.axis2.AxisFault: Transport error: 401 Error: No AuthenticationProvider found
for {0}
at org.apache.axis2.transport.http.HTTPSender.handleResponse(HTTPSender.java:298)
at org.apache.axis2.transport.http.HTTPSender.sendViaPost(HTTPSender.java:192)
at org.apache.axis2.transport.http.HTTPSender.send(HTTPSender.java:77)
at org.apache.axis2.transport.http.CommonsHTTPTransportSender.writeMessageWithCommons(
  CommonsHTTPTransportSender.java:327)
at org.apache.axis2.transport.http.CommonsHTTPTransportSender.invoke(
  CommonsHTTPTransportSender.java:206)
at org.apache.axis2.engine.AxisEngine.send(AxisEngine.java:396)
. . .
```

## Incorrect NNMi Login Configuration

```
Sep 1, 2008 5:13:18.203 PM [ThreadID:1] ALL: com.hp.ov.nms.admin.log.NnmOmFileHandler
init: Logging Handler initialized to log to file pattern: "C:/Documents and Settings/All
Users/Application Data/HP/HP BTO Software\log\nnm\nnm-om.%g.%u.log" up to a limit of 5
files of 1 MB each. Initially handling loggers:"com.hp.ov.nnm.om".

Sep 1, 2008 5:21:40.343 PM [ThreadID:60] CONFIG: com.hp.ov.nnm.om.im.NnmOmModule init:
Initialized nnm: http://nnmserver.domain.com:80 with filter(/sys:onNotification/
arg0[(name='NodeDown' or name='InterfaceDown' or name='SNMPLinkDown')])

Sep 1, 2008 5:21:40.390 PM [ThreadID:60] CONFIG: com.hp.ov.nnm.om.im.NnmOmModule init:
Initialized om: http://omserver.domain.com:8081

Sep 1, 2008 5:21:40.390 PM [ThreadID:60] FINE: com.hp.ov.nnm.om.im.NnmOmModule$OmServer
setSubscription: OM unsubscribing from: omserver.domain.com

Sep 1, 2008 5:21:40.390 PM [ThreadID:60] CONFIG: com.hp.ov.nnm.om.im.NnmOmModule disable:
NNM-OM module disabled

Sep 1, 2008 5:21:55.546 PM [ThreadID:60] SEVERE: com.hp.ov.nnm.om.im.NnmOmModule enable:
Error subscribing:  : javax.xml.ws.WebServiceException: java.io.IOException: Could not
transmit message
at org.jboss.ws.core.jaxws.client.ClientImpl.handleRemoteException(ClientImpl.java:304)
at org.jboss.ws.core.jaxws.client.ClientImpl.invoke(ClientImpl.java:242)
at org.jboss.ws.core.jaxws.client.ClientProxy.invoke(ClientProxy.java:164)
at org.jboss.ws.core.jaxws.client.ClientProxy.invoke(ClientProxy.java:150)
at $Proxy569.subscribeOp(Unknown Source)
at com.hp.ov.nnm.om.im.NnmOmModule.doSubscribe(NnmOmModule.java:1129)
at com.hp.ov.nnm.om.im.NnmOmModule.subscribeNnm(NnmOmModule.java:922)
at com.hp.ov.nnm.om.im.NnmOmModule.enable(NnmOmModule.java:309)
at com.hp.ov.nnm.om.im.NnmOmModule$Enabler$EnableTask.run(NnmOmModule.java:1595)
at java.util.TimerThread.mainLoop(Timer.java:512)
at java.util.TimerThread.run(Timer.java:462)

Caused by: java.io.IOException: Could not transmit message
at org.jboss.ws.core.client.RemotingConnectionImpl.invoke(RemotingConnectionImpl.java:192)
at org.jboss.ws.core.client.SOAPRemotingConnection.invoke(SOAPRemotingConnection.java:77)
at org.jboss.ws.core.CommonClient.invoke(CommonClient.java:331)
at org.jboss.ws.core.jaxws.client.ClientImpl.invoke(ClientImpl.java:230)
. . .

Caused by: org.jboss.remoting.CannotConnectException: Can not connect http client invoker.
```

```
at org.jboss.remoting.transport.http.HTTPClientInvoker.useHttpURLConnection(
  HTTPClientInvoker.java:333)
at org.jboss.remoting.transport.http.HTTPClientInvoker.transport(
  HTTPClientInvoker.java:135)
at org.jboss.remoting.MicroRemoteClientInvoker.invoke(MicroRemoteClientInvoker.java:122)
at org.jboss.remoting.Client.invoke(Client.java:1634)
at org.jboss.remoting.Client.invoke(Client.java:548)
at org.jboss.ws.core.client.RemotingConnectionImpl.invoke(RemotingConnectionImpl.java:171)
. . .

Caused by: org.jboss.ws.WSException: Invalid HTTP server response [401] - Unauthorized
at org.jboss.ws.core.soap.SOAPMessageUnMarshaller.read(SOAPMessageUnMarshaller.java:72)
at org.jboss.remoting.transport.http.HTTPClientInvoker.readResponse(
  HTTPClientInvoker.java:473)
at org.jboss.remoting.transport.http.HTTPClientInvoker.useHttpURLConnection(
  HTTPClientInvoker.java:305)
. . .

Sep 1, 2008 5:21:55.546 PM [ThreadID:60] CONFIG: com.hp.ov.nnm.om.im.NnmOmModule enable:
NNM-OM module enabled
```

## Acknowledgement Error from HPOM for Windows

The following SEVERE log message occurs when an HPOM operator acknowledges a forwarded NNMi incident that has already been closed on the NNMi management server. You can safely ignore this error condition.

```
Sep 2, 2008 12:46:37.234 PM [ThreadID:91] FINE: com.hp.ov.nnm.om.im.NnmOmModule
onOmNotification: onNotification(omserver.domain.com):
9a687f90-78e0-71dd-0713-1019f86f0000/closed

Sep 2, 2008 12:46:40.625 PM [ThreadID:91] FINE: com.hp.ov.nnm.om.im.NnmOmModule
synchronizeIncidentLifecycleState: Synchronizing NNM incident lifecycleState:
9a687f90-78e0-71dd-0713-1019f86f0000/com.hp.nms.incident.lifecycle.Closed

Sep 2, 2008 12:46:45.984 PM [ThreadID:64] FINE: com.hp.ov.nnm.om.im.NnmOmEventSinkBean
onNotification: NnmOmEventSinkBean.onNotification: <id>450971566080</
id><uuid>9a687f90-78e0-71dd-0713-1019f86f0000</uuid><name>SNMPTest</
name><sourceUuid>null</sourceUuid><sourceName>none</sourceName><sourceNodeUuid>null</
sourceNodeUuid><sourceNodeName>16.26.120.61</
sourceNodeName><lifecycleState>com.hp.nms.incident.lifecycle.Closed</
lifecycleState><severity>MINOR</severity><priority>com.hp.nms.incident.priority.None</
priority><category>com.hp.nms.incident.category.Status</
category><family>com.hp.nms.incident.family.Node</family><nature>ROOTCAUSE</
nature><origin>SNMPTRAP</origin><duplicateCount>0</
duplicateCount><priority>com.hp.nms.incident.priority.None</priority><rcaActive>false</
rcaActive><formattedMessage><![CDATA[Received SNMP test packet]]></
formattedMessage><originOccurrenceTime>2008-09-02T12:16:36.000+01:00</
originOccurrenceTime><firstOccurrenceTime>2008-09-02T12:16:39.343+01:00</
firstOccurrenceTime><lastOccurrenceTime>2008-09-02T12:16:39.343+01:00</
lastOccurrenceTime><updateTime>2008-09-02T12:46:43.000+01:00</
updateTime><previousLifecycleState>com.hp.nms.incident.lifecycle.Registered</
previousLifecycleState><previousRcaActive>false</
previousRcaActive><cias><name>cia.snmpoid</name><type>STRING</
type><value>.1.3.6.1.4.1.999.1.0.1001</value></cias><cias><name>cia.address</
name><type>STRING</type><value>16.26.120.61</value></
cias><cias><name>nnmOmEPR.omserver.domain.com</name><type>STRING</type><value>https://
omserver.domain.com:8444/opr-webservice/Incident.svc/
70f54e8c-78df-71dd-1875-1019f8490000</value></cias><cias><name>.1.3</
name><type>ASN_INTEGER</type><value>100</value></cias>

Sep 2, 2008 12:46:46.000 PM [ThreadID:64] FINE: com.hp.ov.nnm.om.im.NnmOmModule
onIncidentUpdate: Incident update received: 9a687f90-78e0-71dd-0713-1019f86f0000
<id>450971566080</id><uuid>9a687f90-78e0-71dd-0713-1019f86f0000</uuid><name>SNMPTest</
name><sourceUuid>null</sourceUuid><sourceName>none</sourceName><sourceNodeUuid>null</
```

sourceNodeUuid><sourceNodeName>16.26.120.61</
sourceNodeName><lifecycleState>com.hp.nms.incident.lifecycle.Closed</
lifecycleState><severity>MINOR</severity><priority>com.hp.nms.incident.priority.None</
priority><category>com.hp.nms.incident.category.Status</
category><family>com.hp.nms.incident.family.Node</family><nature>ROOTCAUSE</
nature><origin>SNMPTRAP</origin><duplicateCount>0</
duplicateCount><priority>com.hp.nms.incident.priority.None</priority><rcaActive>false</
rcaActive><formattedMessage><![CDATA[Received SNMP test packet]]></
formattedMessage><originOccurrenceTime>2008-09-02T12:16:36.000+01:00</
originOccurrenceTime><firstOccurrenceTime>2008-09-02T12:16:39.343+01:00</
firstOccurrenceTime><lastOccurrenceTime>2008-09-02T12:16:39.343+01:00</
lastOccurrenceTime><updateTime>2008-09-02T12:46:43.000+01:00</
updateTime><previousLifecycleState>com.hp.nms.incident.lifecycle.Registered</
previousLifecycleState><previousRcaActive>false</
previousRcaActive><cias><name>cia.snmpoid</name><type>STRING</
type><value>.1.3.6.1.4.1.999.1.0.1001</value></cias><cias><name>cia.address</
name><type>STRING</type><value>16.26.120.61</value></
cias><cias><name>nnmOmEPR.omserver.domain.com</name><type>STRING</type><value>https://
omserver.domain.com:8444/opr-webservice/Incident.svc/
70f54e8c-78df-71dd-1875-1019f8490000</value></cias><cias><name>.1.3</
name><type>ASN_INTEGER</type><value>100</value></cias>

```
Sep 2, 2008 12:46:46.031 PM [ThreadID:64] FINE: com.hp.ov.nnm.om.im.NnmOmModule
ackNnmIncidentToOm: Acknowledging incident 9a687f90-78e0-71dd-0713-1019f86f0000
omIncident(70f54e8c-78df-71dd-1875-1019f8490000) to: omserver.domain.com
```

**Sep 2, 2008 12:46:47.078 PM [ThreadID:64] SEVERE: com.hp.ov.nnm.om.im.OmBean acknowledgeIncident: Om acknowledge failure:  : org.apache.axis2.AxisFault: The service cannot comply with the request due to internal processing errors.**

```
at org.apache.axis2.util.Utils.getInboundFaultFromMessageContext(Utils.java:486)
at
org.apache.axis2.description.OutInAxisOperationClient.handleResponse(OutInAxisOperation.ja
va:343)
at org.apache.axis2.description.OutInAxisOperationClient.send(OutInAxisOperation.java:389)
at
org.apache.axis2.description.OutInAxisOperationClient.executeImpl(OutInAxisOperation.java:
211)
at org.apache.axis2.client.OperationClient.execute(OperationClient.java:163)
at com.hp.ov.opr.ws.client.axis.IncidentServiceStub.Close(IncidentServiceStub.java:756)
at com.hp.ov.opr.ws.client.axis.OmWsClient.closeOp(OmWsClient.java:411)
at com.hp.ov.nnm.om.im.OmBean.acknowledgeIncident(OmBean.java:194)
. . .
```

## Acknowledgement Error from HPOM for UNIX

The following SEVERE log message occurs when an HPOM operator acknowledges a forwarded NNMi incident that has already been closed on the NNMi management server. You can safely ignore this error condition.

```
Sep 3, 2008 12:53:23.728 PM [ThreadID:119] FINE: com.hp.ov.nnm.om.im.NnmOmModule
onOmNotification: onNotification(omserver.domain.com):
77f7bab0-7963-71dd-0142-ac1107ce0000/open

Sep 3, 2008 12:54:22.307 PM [ThreadID:121] FINE: com.hp.ov.nnm.om.im.NnmOmModule
onOmNotification: onNotification(omserver.domain.com):
77f7bab0-7963-71dd-0142-ac1107ce0000/closed

Sep 3, 2008 12:54:22.589 PM [ThreadID:121] FINE: com.hp.ov.nnm.om.im.NnmOmModule
synchronizeIncidentLifecycleState: Synchronizing NNM incident lifecycleState:
77f7bab0-7963-71dd-0142-ac1107ce0000/com.hp.nms.incident.lifecycle.Closed

Sep 3, 2008 12:54:23.089 PM [ThreadID:120] FINE: com.hp.ov.nnm.om.im.NnmOmEventSinkBean
onNotification: NnmOmEventSinkBean.onNotification: <id>1404454305792</
id><uuid>77f7bab0-7963-71dd-0142-ac1107ce0000</uuid><name>SNMPColdStart</
name><sourceUuid>null</sourceUuid><sourceName>none</
```

sourceName><sourceNodeUuid>3d52aff2-eb9d-45de-8e66-3b882fc6e533</
sourceNodeUuid><sourceNodeName>chinnm72</
sourceNodeName><lifecycleState>com.hp.nms.incident.lifecycle.Closed</
lifecycleState><severity>NORMAL</severity><priority>com.hp.nms.incident.priority.None</
priority><category>com.hp.nms.incident.category.Fault</
category><family>com.hp.nms.incident.family.Node</family><nature>SYMPTOM</
nature><origin>SNMPTRAP</origin><duplicateCount>0</
duplicateCount><priority>com.hp.nms.incident.priority.None</priority><rcaActive>false</
rcaActive><formattedMessage><![CDATA[Agent Up with Possible Changes (coldStart Trap)]]></
formattedMessage><originOccurrenceTime>2008-09-03T12:53:21.000+10:00</
originOccurrenceTime><firstOccurrenceTime>2008-09-03T12:53:21.259+10:00</
firstOccurrenceTime><lastOccurrenceTime>2008-09-03T12:53:21.259+10:00</
lastOccurrenceTime><updateTime>2008-09-03T12:54:22.964+10:00</
updateTime><previousLifecycleState>com.hp.nms.incident.lifecycle.Registered</
previousLifecycleState><previousRcaActive>false</
previousRcaActive><cias><name>cia.snmpoid</name><type>STRING</
type><value>.1.3.6.1.6.3.1.1.5.1</value></cias><cias><name>cia.address</
name><type>STRING</type><value>172.17.7.206</value></
cias><cias><name>nnmOmEPR.omserver.domain.com</name><type>STRING</type><value>https://
chiomw71.simnetad.simplot.com.au/opr-webservice/Incident.svc/
833cd1a2-630b-4ba1-956a-b5450bd15dce</value></cias>

Sep 3, 2008 12:54:23.089 PM [ThreadID:120] FINE: com.hp.ov.nnm.om.im.NnmOmModule
onIncidentUpdate: Incident update received: 77f7bab0-7963-71dd-0142-ac1107ce0000
<id>1404454305792</id><uuid>77f7bab0-7963-71dd-0142-ac1107ce0000</
uuid><name>SNMPColdStart</name><sourceUuid>null</sourceUuid><sourceName>none</
sourceName><sourceNodeUuid>3d52aff2-eb9d-45de-8e66-3b882fc6e533</
sourceNodeUuid><sourceNodeName>chinnm72</
sourceNodeName><lifecycleState>com.hp.nms.incident.lifecycle.Closed</
lifecycleState><severity>NORMAL</severity><priority>com.hp.nms.incident.priority.None</
priority><category>com.hp.nms.incident.category.Fault</
category><family>com.hp.nms.incident.family.Node</family><nature>SYMPTOM</
nature><origin>SNMPTRAP</origin><duplicateCount>0</
duplicateCount><priority>com.hp.nms.incident.priority.None</priority><rcaActive>false</
rcaActive><formattedMessage><![CDATA[Agent Up with Possible Changes (coldStart Trap)]]></
formattedMessage><originOccurrenceTime>2008-09-03T12:53:21.000+10:00</
originOccurrenceTime><firstOccurrenceTime>2008-09-03T12:53:21.259+10:00</
firstOccurrenceTime><lastOccurrenceTime>2008-09-03T12:53:21.259+10:00</
lastOccurrenceTime><updateTime>2008-09-03T12:54:22.964+10:00</
updateTime><previousLifecycleState>com.hp.nms.incident.lifecycle.Registered</
previousLifecycleState><previousRcaActive>false</
previousRcaActive><cias><name>cia.snmpoid</name><type>STRING</
type><value>.1.3.6.1.6.3.1.1.5.1</value></cias><cias><name>cia.address</
name><type>STRING</type><value>172.17.7.206</value></
cias><cias><name>nnmOmEPR.omserver.domain.com</name><type>STRING</type><value>https://
chiomw71.simnetad.simplot.com.au/opr-webservice/Incident.svc/
833cd1a2-630b-4ba1-956a-b5450bd15dce</value></cias>

Sep 3, 2008 12:54:23.104 PM [ThreadID:120] FINE: com.hp.ov.nnm.om.im.NnmOmModule
ackNnmIncidentToOm: Acknowledging incident 77f7bab0-7963-71dd-0142-ac1107ce0000
omIncident(833cd1a2-630b-4ba1-956a-b5450bd15dce) to: omserver.domain.com

**Sep 3, 2008 12:54:23.229 PM [ThreadID:120] SEVERE: com.hp.ov.nnm.om.im.OmBean acknowledgeIncident: Om acknowledge failure: : org.apache.axis2.AxisFault: The destination represented by the WS-Addressing To EPR with address Unable to find specified incident 833cd1a2-630b-4ba1-956a-b5450bd15dce**
at omwWS.DataAccess.WmiProxy.executeMassCommand(Command cmd, List`1 incidentIdList) in
S:\DE\oprweb\om-ws\server\omw\Source\Resource Access\omwWS.DataAccess\WmiProxy.cs:line 576
at omwWS.ServiceImplementation.Incident.Close(CloseRequest request) in
S:\DE\oprweb\om-ws\server\omw\Source\Service
Interface\omwWS.ServiceImplementation\Incident.cs:line 1080 cannot be reached.
at
org.apache.axis2.handlers.addressing.AddressingInFaultHandler.invoke(AddressingInFaultHand
ler.java:114)
at org.apache.axis2.engine.Phase.invoke(Phase.java:292)
at org.apache.axis2.engine.AxisEngine.invoke(AxisEngine.java:212)
at org.apache.axis2.engine.AxisEngine.receive(AxisEngine.java:132)

```
at
org.apache.axis2.description.OutInAxisOperationClient.handleResponse(OutInAxisOperation.ja
va:336)
at org.apache.axis2.description.OutInAxisOperationClient.send(OutInAxisOperation.java:389)
at
org.apache.axis2.description.OutInAxisOperationClient.executeImpl(OutInAxisOperation.java:
211)
at org.apache.axis2.client.OperationClient.execute(OperationClient.java:163)
at com.hp.ov.opr.ws.client.axis.IncidentServiceStub.Close(IncidentServiceStub.java:756)
at com.hp.ov.opr.ws.client.axis.OmWsClient.closeOp(OmWsClient.java:411)
at com.hp.ov.nnm.om.im.OmBean.acknowledgeIncident(OmBean.java:194)
. . .
```

## Node Not Found Error

HPOM does not recognize the source of the forwarded NNMi incident because the source is not correctly configured as on HPOM managed node.

```
Sep 9, 2008 8:40:51.796 AM [ThreadID:75] SEVERE: com.hp.ov.nnm.om.im.NnmOmModule getNode:
Error retrieving node: null : com.hp.ov.nms.sdk.node.NmsNodeException: Cannot interrogate
model:
at sun.reflect.NativeConstructorAccessorImpl.newInstance0(Native Method)
at
sun.reflect.NativeConstructorAccessorImpl.newInstance(NativeConstructorAccessorImpl.java:3
9)
at
sun.reflect.DelegatingConstructorAccessorImpl.newInstance(DelegatingConstructorAccessorImp
l.java:27)
at java.lang.reflect.Constructor.newInstance(Constructor.java:494)
at org.jboss.ws.metadata.umdm.FaultMetaData.toServiceException(FaultMetaData.java:384)
at
org.jboss.ws.core.jaxws.SOAPFaultHelperJAXWS.getSOAPFaultException(SOAPFaultHelperJAXWS.ja
va:129)
at
org.jboss.ws.core.jaxws.binding.SOAP11BindingJAXWS.throwFaultException(SOAP11BindingJAXWS.
java:109)
at org.jboss.ws.core.CommonSOAPBinding.unbindResponseMessage(CommonSOAPBinding.java:567)
at org.jboss.ws.core.CommonClient.invoke(CommonClient.java:365)
at org.jboss.ws.core.jaxws.client.ClientImpl.invoke(ClientImpl.java:230)
at org.jboss.ws.core.jaxws.client.ClientProxy.invoke(ClientProxy.java:164)
at org.jboss.ws.core.jaxws.client.ClientProxy.invoke(ClientProxy.java:150)
at $Proxy583.getNodes(Unknown Source)
. . .
```

# HP Universal Configuration Management Database

HP Universal Configuration Management Database (UCMDB) automatically maintains accurate, up-to-date information on infrastructure and application relationships through native integration to HP Discovery and Dependency mapping (DDM). UCMDB is beneficial for the following tasks:

- Using impact modeling to show the rippling effect of infrastructure and application changes before they occur.
- Tracking actual planned and unplanned changes through discovered change history.
- Gaining a shared, authoritative view of the environment through awareness of existing data repositories.

For information about purchasing UCMDB, contact your HP sales representative.

This chapter contains the following topics:


- HP NNMi–HP UCMDB Integration
- Enabling the HP NNMi–HP UCMDB Integration
- Using the HP NNMi–HP UCMDB Integration
- Changing the HP NNMi–HP UCMDB Integration Configuration
- Disabling the HP NNMi–HP UCMDB Integration
- Troubleshooting the HP NNMi–HP UCMDB Integration
- HP NNMi–HP UCMDB Integration Configuration Form Reference


## HP NNMi–HP UCMDB Integration

The HP NNMi–HP UCMDB integration shares NNMi topology information with UCMDB. UCMDB stores each device in the NNMi topology as a configuration item (CI). UCMDB applies Discovery and Dependency Mapping (DDM) patterns to the CIs for the NNMi topology to predict the impact of a device failure. This impact analysis is available from the UCMDB user interface and also from the NNMi console.

## Value

The HP NNMi–HP UCMDB integration allows NNMi to be the authoritative source for network device relationships.

## Supported Versions

The information in this chapter applies to the following product versions:

- UCMDB version 8.00 or higher
- NNMi version 8.10 or higher

NNMi and UCMDB cannot be installed on the same computer. The two products must be installed on different computers in either of the following configurations:

- Different operating systems. For example, the NNMi management server is a Linux system, and the UCMDB server is a Windows system.
- The same operating system. For example, the NNMi management server is a Windows system, and the UCMDB server is a second Windows system.

For the most recent information about supported hardware platforms and operating systems, refer to the support matrices for both products.

## Documentation

This chapter describes how to configure NNMi to communicate with UCMDB and how to use the integration from the NNMi console.

The *HP Universal CMDB–HP Network Node Manager (NNMi) Integration Guide*, which is included on the UCMDB product media, describes how to configure UCMDB to communicate with NNMi. It also describes how to use the integration from the UCMDB user interface.

# Enabling the HP NNMi–HP UCMDB Integration

1. On the UCMDB server, configure the connection between UCMDB and NNMi, and customize the integration:
   a. Deploy the UCMDB–NNMi Integration Discovery package.
   b. Set up and start the DDM probe for NNMi topology data.
   c. Set up the Network – NNM Layer2 jobs.

   For more information, see the *HP Universal CMDB–HP Network Node Manager (NNMi) Integration Guide*.
2. On the NNMi management server, configure the connection between NNMi and UCMDB:
   a. In the NNMi console, open the **HP NNMi–HP UCMDB Integration Configuration** form (**Integration Module Configuration > HP UCMDB**).
   b. Select the **Enable Integration** check box to activate the remaining fields in the form.

c Enter the information for connecting to the NNMi management server.

For information about these fields, see NNMi Management Server Connection on page 218.

d Enter the information for connecting to the UCMDB server.

For information about these fields, see UCMDB Server Connection on page 219.

e Click **Submit** at the bottom of the form.

A new window displays a status message. If the message indicates a problem with connecting to the UCMDB server, re-open the **HP NNMi–HP UCMDB Integration Configuration** form (or press **ALT+LEFT ARROW** in the message window), and then adjust the values for connecting to the UCMDB server as suggested by the text of the error message.

3 On the NNMi management server, customize the integration:

a In the NNMi console, open the **HP NNMi–HP UCMDB Integration Configuration** form (**Integration Module Configuration > HP UCMDB**).

b Enter values for the following fields:

— **HP UCMDB Correlation Rule Prefix**

— **HP UCMDB Impact Severity Level (1–9)**

For information about these fields, see Integration Behavior on page 219.

c Click **Submit** at the bottom of the form.

# Using the HP NNMi–HP UCMDB Integration

Enabling the HP NNMi–HP UCMDB integration adds two URL actions to the NNMi console:

- The **Find UCMDB Impacted CIs** action, which is described in Viewing Impacted CIs.
- The **Open CI in UCMDB** action, which is described in Viewing the UCMDB CI.

For information about using the integration from the UCMDB user interface, see the *HP Universal CMDB–HP Network Node Manager (NNMi) Integration Guide*.

## Viewing Impacted CIs

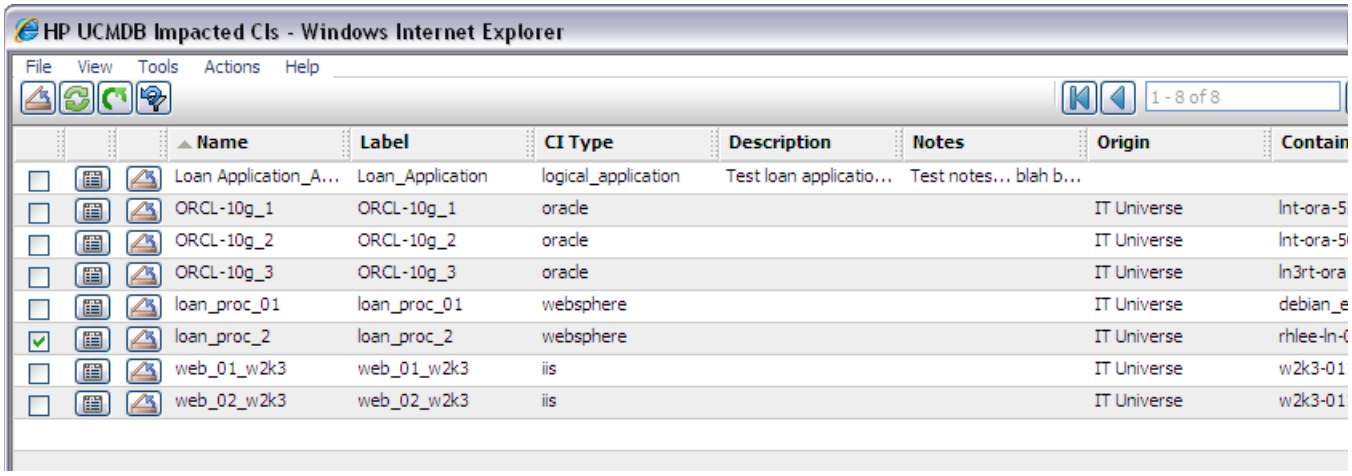
The **Find UCMDB Impacted CIs** action displays a list of the UCMDB configuration items that would be impacted (according to the correlation rules and severity level configured for the integration) for the selected node or interface. This action is available from the following NNMi console locations:

- Any node inventory view
- Any interface inventory view
- Any map view (with a node or interface selected)
- Any incident browser

The **Find UCMDB Impacted CIs** action is available for all nodes and interfaces in the NNMi topology, regardless of whether these objects are modeled in the UCMDB database.

Figure 14 shows example results from the **Find UCMDB Impacted CIs** action.

**Figure 14 Example Impacted CIs Table**




## Viewing the UCMDB CI

The **Open CI in UCMDB** action displays the UCMDB information about the selected CI.

# Changing the HP NNMi–HP UCMDB Integration Configuration

1. In the NNMi console, open the **HP NNMi–HP UCMDB Integration Configuration** form (**Integration Module Configuration > HP UCMDB**).
2. Modify the values as appropriate.

   For information about the fields on this form, see HP NNMi–HP UCMDB Integration Configuration Form Reference on page 217.
3. Verify that the **Enable Integration** check box at the top of the form is selected, and then click **Submit** at the bottom of the form.

The changes take effect immediately. You do not need to restart `ovjboss`.

# Disabling the HP NNMi–HP UCMDB Integration

1. In the NNMi console, open the **HP NNMi–HP UCMDB Integration Configuration** form (**Integration Module Configuration > HP UCMDB**).
2. Clear the **Enable Integration** check box at the top of the form, and then click **Submit** at the bottom of the form.

   The integration URL actions are no longer available.

➤ The changes take effect immediately. You do not need to restart `ovjboss`.

# Troubleshooting the HP NNMi–HP UCMDB Integration

Troubleshooting information will be added as it becomes available.

For information about retrieving an updated version of this document, see Available Product Documentation on page 3.

# HP NNMi–HP UCMDB Integration Configuration Form Reference

The **HP NNMi–HP UCMDB Integration Configuration** form contains the parameters for communications between NNMi and UCMDB. This form is available from the **Integration Module Configuration** workspace.

➤ Only NNMi users with the Administrator role can access the **HP NNMi–HP UCMDB Integration Configuration** form.

The **HP NNMi–HP UCMDB Integration Configuration** form collects information for the following general areas:

- NNMi Management Server Connection
- UCMDB Server Connection
- Integration Behavior

To apply changes to the integration configuration, update the values on the **HP NNMi–HP UCMDB Integration Configuration** form, and then click **Submit**.

## NNMi Management Server Connection

Table 19 lists the parameters for connecting to the NNMi management server. This is the same information that you use to open the NNMi console. You can determine many of these values by examining the URL that invokes an NNMi console session. Coordinate with the NNMi administrator to determine the appropriate values for this section of the configuration form.

The default NNMi configuration uses http for connecting to the NNMi console. For information about configuring this connection to use https, see Enabling https for NNMi on page 73.

**Table 19 NNMi Management Server Information**

| Field | Description |
|---|---|
| NNMi SSL Enabled | The connection protocol specification.<br>• If the NNMi console is configured to use https, select the **NNMi SSL Enabled** check box.<br>• If the NNMi console is configured to use http, clear the **NNMi SSL Enabled** check box. This is the default configuration. |
| NNMi Host | The fully-qualified domain name of the NNMi management server. This field is pre-filled with host name that was used to access the NNMi console. Verify that this value is the name that is returned by the **`nnmofficialfqdn.ovpl -t`** command run on the NNMi management server. |
| NNMi Port | The port for connecting to the NNMi console. This field is pre-filled with the port that the jboss application server uses for communicating with the NNMi console, as specified in the following file:<br>• *Windows*: `%NnmDataDir%\shared\nnm\conf\nnm.ports.properties`<br>• *UNIX*: `$NnmDataDir/shared/nnm/conf/nnm.ports.properties`<br>For non-SSL connections, use the value of `jboss.http.port`, which is `80` or `8004` by default (depending on the presence of another web server when NNMi was installed).<br>For SSL connections, use the value of `jboss.https.port`, which is `443` by default. |
| NNMi User | The user name for connecting to the NNMi console. This user must have the NNMi Administrator or Web Service Client role. |
| NNMi Password | The password for the specified NNMi user. |

## UCMDB Server Connection

Table 20 lists the parameters for connecting to the web services on the UCMDB server. Coordinate with the UCMDB administrator to determine the appropriate values for this section of the configuration.

**Table 20 UCMDB Management Server Information**

| HPOM Server Parameter | Description |
|---|---|
| HP UCMDB SSL Enabled | The connection protocol specification for connecting to the UCMDB web services.<br>• If the UCMDB web services are configured to use https, select the **HP UCMDB SSL Enabled** check box. This is the default configuration.<br>• If the UCMDB web services are configured to use http, clear the **HP UCMDB SSL Enabled** check box. |
| HP UCMDB Host | The fully-qualified domain name of the UCMDB server. |
| HP UCMDB Port | The port for connecting to the UCMDB web services.<br>If you are using the default UCMDB configuration, use port `8080` (for SSL connections to UCMDB). |
| HP UCMDB User | A valid UCMDB user account name with the UCMDB Administrator role. |
| HP UCMDB Password | The password for the specified UCMDB user. |

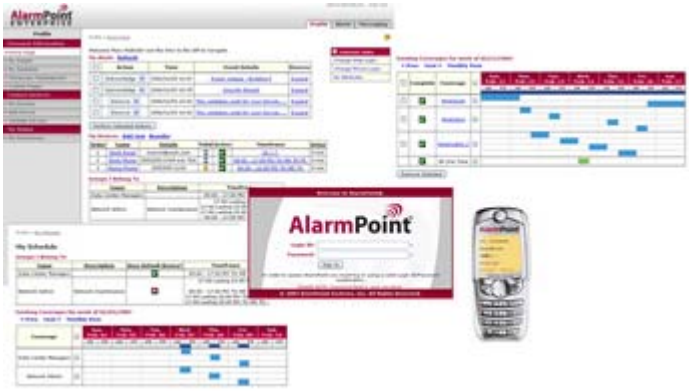
## Integration Behavior

Table 21 lists the parameters that describe the integration behavior. Coordinate with the UCMDB administrator to determine the appropriate values for this section of the configuration.

**Table 21 Integration Behavior Information**

| Field | Description |
|---|---|
| HP UCMDB Correlation Rule Prefix | The prefix of the UCMDB impact correlation rules that apply to NNMi topology objects. |
| HP UCMDB Impact Severity Level (1–9) | The severity level at which to apply the UCMDB impact correlation rules. HP recommends using the highest severity, 9, to include all rules in the calculation of possible impact. |

# AlarmPoint




AlarmPoint is an interactive alerting application, designed to capture and enrich important events, to route those events to the correct person on any communication device, and to give that person the ability to solve, escalate, or enlist others to resolve the situation.

Through integrations, the AlarmPoint System can become the voice and interface of an automation engine or an intelligent application, such as HP Network Node Manager i Software. When NNMi detects an event that requires attention, AlarmPoint places phone calls or sends pages, instant messages, or e-mail messages to the appropriate personnel, vendors, or customers.

The AlarmPoint System is also persistent, escalating through multiple devices and personnel until someone accepts responsibility or resolves the event. The AlarmPoint System gives the notified person instant, two-way communication with NNMi. Responses are processed immediately on the NNMi management server, enabling remote resolution of the event.

For information about purchasing AlarmPoint, contact your HP sales representative.

This chapter contains the following topics:


- HP NNMi–AlarmPoint Integration
- Enabling the HP NNMi–AlarmPoint Integration
- Using the HP NNMi–AlarmPoint Integration
- Disabling the HP NNMi–AlarmPoint Integration
- Troubleshooting the HP NNMi–AlarmPoint Integration


## HP NNMi–AlarmPoint Integration

By including AlarmPoint in the HP Network Node Manager i-series Software environment, network administrators who use NNMi to monitor and manage their network devices gain intelligent alerting and two-way communication between personal communication tools and NNMi.

The HP NNMi–AlarmPoint integration supports event notifications (from NNMi to AlarmPoint) through the configuration of NNMi incident types. It also supports inbound actions (from AlarmPoint to NNMi) to acknowledge the original incident, alter its priority, and add informational annotations.

## Value

With the HP NNMi–AlarmPoint integration, the appropriate technician can be notified directly through voice, e-mail, pager, or another device. The event resolver receives information about the failure and is able to make decisions in real time, such as acknowledging, ignoring, annotating, or changing the priority of the event.

After the recipient selects a response on their remote device, AlarmPoint updates the NNMi incident in real time. The benefit is that this process is immediate—significantly faster than the time required for operations staff to notice the failures or malfunctions, determine who is on call, and then manually notify the correct person. Being able to take simple actions to update the incident from any device gives the event resolver a quick way to deal with many issues and communicate to other team members the current status of the incident.

During the process, AlarmPoint logs every notification, response, and action. In addition, AlarmPoint automatically annotates the original NNMi incident with status information.

The AlarmPoint product features a self-service web-based user interface for assigning responsible personnel to each job. AlarmPoint also includes an optional enhanced Subscription panel that allows both managed and self subscription to NNMi incidents.

AlarmPoint Express, a limited edition of AlarmPoint, is available for select versions of NNMi. AlarmPoint Express for HP NNMi has a reduced feature set and comes pre-configured for an NNMi integration. AlarmPoint Express is a great way to get started with the AlarmPoint product family, or it is well-suited for a small production environment that does not require voice or distributed load capability.

## Supported Versions

The information in this chapter applies to the following product versions:

- AlarmPoint Java Client version 3.2.1 or higher
- AlarmPoint version 3.2.1 or higher
- NNMi version 8.01 or higher, with an NNMi integration enablement license

It is recommended that NNMi and AlarmPoint be installed on different computers. The NNMi management server and the AlarmPoint server can be of the same or different operating systems.

## Documentation

The HP NNMi–AlarmPoint integration is fully described in the *AlarmPoint for HP Network Node Manager i-series Integration Guide*, which is included with the integration.

The AlarmPoint documentation suite describes the AlarmPoint features and capabilities in detail. The documentation suite is available for download from the AlarmPoint Customer Support Site at:

**http://support.alarmpoint.com/**

The *AlarmPoint Express for HP NNMi Quick Start Guide* provides an introduction to AlarmPoint's features. This guide describes how to install, configure, and maintain an HP NNMi–AlarmPoint Express integration.

# Enabling the HP NNMi–AlarmPoint Integration

The high-level steps for installing and configuring the AlarmPoint for NNMi integration are as follows. For detailed information, refer to the *AlarmPoint for HP Network Node Manager i-series Integration Guide*.

1. Install the AlarmPoint Java Client on the NNMi management server.
2. Install the NNMi-specific integration script for the AlarmPoint Java Client.
3. Install the Web Services Library on the AlarmPoint Webservers and the Application server.
4. *Optional*. Install the AlarmPoint subscription panel for NNMi on the AlarmPoint Webservers.
5. Install the AlarmPoint action scripts for NNMi by using the AlarmPoint Developer IDE.
6. Install the integration voice files to the AlarmPoint Application server.
7. Configure an Event Domain (and, optionally, a Subscription Domain) in AlarmPoint.
8. Configure a Web Services Client for NNMi.
9. Configure the NNMi incident types that should trigger the AlarmPoint scripts.
10. Validate that the integration can inject NNMi incident parameters for AlarmPoint notifications, and that AlarmPoint responses properly update the NNMi incident.

# Using the HP NNMi–AlarmPoint Integration

NNMi and AlarmPoint interact by delivering notifications to users and injecting the responses back into NNMi. When NNMi detects a problem in the network (for example, a NonSNMPNodeUnresponsive incident), the following process occurs:

1. NNMi calls the AlarmPoint Client (APClient) with the parameters describing the problem (for example, the computer affected and the situation).
2. The APClient submits the information to the AlarmPoint Agent (APAgent).
3. The APAgent ensures delivery of the problem details to AlarmPoint, which in turn notifies the appropriate recipient.
4. The recipient responds to the notification. The recipient's acknowledgement, annotation, or priority change updates NNMi through a Web services call.

## Disabling the HP NNMi–AlarmPoint Integration

To disable the integration, remove the components installed by the integration's executable archive.

For information about removing an AlarmPoint deployment, refer to the *AlarmPoint for HP Network Node Manager i-series Integration Guide*.

## Troubleshooting the HP NNMi–AlarmPoint Integration

For information about optimizing and extending the integration, and any currently known issues, refer to the *AlarmPoint for HP Network Node Manager i-series Integration Guide*.

# nGenius Performance Manager

NetScout Systems nGenius Performance Manager provides visibility into complex networks for the following purposes:

- Application recognition and monitoring
- Analysis and troubleshooting of packets and flows
- Response time analysis
- Reporting and capacity planning
- Convergence management
- Alarming and event identification

nGenius Performance Manager leverages deep packet inspection and flow-based technologies to deliver visibility into real-time, operational intelligence that spans several types of data:

- High-level key performance indicators (KPIs), such as response time, errors, or jitter
- Application flow data, such as utilization, conversations, or top talkers
- Packet-level analysis, such as decodes and bounce diagrams

nGenius Performance Manager collects performance data from a wide variety of network data sources, allowing it to monitor the usage patterns of all network infrastructures, topologies, and applications. nGenius Performance Manager then presents the results in a collection of real-time and historical views and reports that can show the following information:

- Application performance
- Users and abusers of network resources
- Resource consumption of network capacity

For information about purchasing nGenius Performance Manager, contact your HP sales representative.

This chapter contains the following topics:


- HP NNMi–nGenius Performance Manager Integration
- Enabling the HP NNMi–nGenius Performance Manager Integration
- Using the HP NNMi–nGenius Performance Manager Integration
- Disabling the HP NNMi–nGenius Performance Manager Integration

# HP NNMi–nGenius Performance Manager Integration

By including nGenius Performance Manager in the HP Network Node Manager i-series Software environment, network administrators who use NNMi to monitor and manage their network devices gain application-level visibility through nGenius devices.

The HP NNMi–nGenius Performance Manager integration provides the following functionality:

- Receive nGenius Performance Manager server and probe alarms in the NNMi incident views.
- Investigate the cause of an incident by launching contextual views into nGenius Performance Manager.
- Display nGenius Probes with a NetScout icon in NNMi map views.
- Launch nGenius Performance Manager QuickViews from the NNMi console.
- Launch the nGenius Performance Manager application from the NNMi console.
- Launch NNMi Layer 2 and Layer 3 Neighbor views for the nGenius Performance Manager incidents that are available in the NNMi incident views.
- Launch the NNMi Path View map for nGenius Performance Manager incidents that are available in the NNMi incident views.
- Forward Clear Trap alarms to NNMi for each alarm generated by nGenius Performance Manager.

## Value

The HP NNMi–nGenius Performance Manager integration provides the following benefits:

- Reduces the cost of delivering maximum network availability.
- Consolidates the network management infrastructure in a single console.
- Increases staff productivity and efficiency with integrated fault and application-aware performance data.
- Shrinks MTTR with contextual drill down to flow and packet-level details to identify the performance problems.

## Supported Versions

The information in this chapter applies to the following product versions:

- nGenius Performance Manager version 4.3 build 450 or higher
- nGenius K2 version 1.0
- CDM Agent Firmware version 4.3 build 130 or higher
- nGenius AFMon appliance version 4.3 build 555 or higher

- nGenius InfiniStream Capture Engine version 4.5 or higher
- NNMi version 8.02 or higher

NNMi and nGenius Performance Manager Server must be installed on different computers. The NNMi management server and the nGenius Performance Manager Server computer can be of the same or different operating systems.

### Documentation

The HP NNMi–nGenius Performance Manager integration is fully described in the following document:

> *Integrating nGenius Performance Manager v4.3 with HP Network Node Manager i Software* (NetScout part number 733-0089 Rev. A, available from **http://www.netscout.com**)

## Enabling the HP NNMi–nGenius Performance Manager Integration

The nGenius Performance Manager Integration utility for HP Network Node Manager installation file is available in the following location on the nGenius Performance Manager server:

- *Windows*: `%nGenius Install%\rtm\bin\nGeniusNNM8.zip`
- *UNIX*: `$nGenius Install/rtm/bin/nGeniusNNM8.zip`

As a user with administrative or root privileges, install the integration utility on the NNMi management server. The utility imports all NNMi integration-related configuration data for the nGenius Server into NNMi.

The high-level steps for installing the nGenius Performance Manager Integration utility are as follows. For detailed information, refer to *Integrating nGenius Performance Manager v4.3 with HP Network Node Manager i Software*.

1. Extract the installation files and configure NNMi support within nGenius Performance Manager.
2. Configure NetScout incidents (alarms) in NNMi.
3. Configure the nGenius Probe to send SNMP traps to port 395.
4. *Optional*. Configure Probe Router Mapping.

## Using the HP NNMi–nGenius Performance Manager Integration

For information about using the HP NNMi–nGenius Performance Manager integration, refer to *Integrating nGenius Performance Manager v4.3 with HP Network Node Manager i Software*.

## Disabling the HP NNMi–nGenius Performance Manager Integration

For information about disabling the HP NNMi–nGenius Performance Manager integration, refer to *Integrating nGenius Performance Manager v4.3 with HP Network Node Manager i Software*.

## Troubleshooting the HP NNMi–nGenius Performance Manager Integration

For information about troubleshooting the HP NNMi–nGenius Performance Manager integration, contact NetScout Systems Customer Support. Contact information is available at:

**http://www.netscout.com/support**

# NNM iSPI for Performance

This chapter describes considerations and best practices for integrating the NNM iSPI for Performance with NNMi.

For information about purchasing the NNM iSPI for Performance, contact your HP sales representative.

This chapter contains the following topics:


- What is the NNM iSPI for Performance?
- Enabling the NNM iSPI for Performance
- Configuring the NNM iSPI for Performance to Run Under HA
- Using the NNM iSPI for Performance
- Backup Provisions for the NNM iSPI for Performance
- Resetting the NNM iSPI for Performance
- Moving the NNM iSPI for Performance to a Different Computer
- Disabling the NNM iSPI for Performance
- Troubleshooting the NNM iSPI for Performance


## What is the NNM iSPI for Performance?

The NNM iSPI for Performance enables data collection, threshold monitoring, and threshold alarming for a predefined set of MIB variables within HP Network Node Manager i Software. From NNMi, you can launch NNM iSPI for Performance reports.

The NNMi performance monitoring settings determine the interfaces for which NNMi collects performance data and the intervals of the data collection. The NNM iSPI for Performance receives and analyzes the NNMi data and then stores the analysis results in a proprietary format.

The NNM iSPI for Performance is licensed separately from NNMi; however, both products must be licensed for the same node count.

## Value

NNM iSPI for Performance adds performance reporting to NNMi for ranking, trending, and exception reporting.

- Ranking sorts elements by utilization, and highlights network elements with unusually high or unusually low utilization.
- Trending looks at a period of time and shows, through color-coding, whether performance is stable, worsening, or improving.
- Exception reporting looks at threshold conditions, and tracks threshold conditions over time.

## Supported Versions

The information in this chapter applies to the following product versions:

- NNM iSPI for Performance version 8.10 or higher
- NNMi version 8.10 or higher

## Documentation

The NNM iSPI for Performance release notes, support matrix, and installation instructions are available from:

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

Additionally, NNM iSPI for Performance help is installed with the product.

# Enabling the NNM iSPI for Performance

The NNM iSPI for Performance can be installed as an add-on product on the NNMi management server or on a dedicated server. The recommended installation model depends on the frequency and volume of performance polling. For more information, see the NNM iSPI for Performance support matrix.

As you plan the installation, remember to add the disk space requirements and to check the RAM requirements for all software that you plan to install on a single computer.

The following steps present a high-level process for enabling the NNM iSPI for Performance:

1. On each NNMi management server, run the NNM iSPI for Performance enablement script.

   For detailed information, see the *NNM iSPI for Performance Installation Guide*.

2. Install the NNM iSPI for Performance as described in the *NNM iSPI for Performance Installation Guide*.

3. In the NNMi console, configure performance monitoring.

   For information, see *Setting Polling Characteristics in NNMi* and *Setting Thresholds for Metrics* in the NNM iSPI for Performance help.

4 In the NNM iSPI for Performance, configure performance reporting.

For information, see *Scheduling Reports* in the NNM iSPI for Performance help.

# Configuring the NNM iSPI for Performance to Run Under HA

If you plan to run NNMi under high availability (HA), the following options are available:

- If the NNM iSPI for Performance is installed on the NNMi management servers, the iSPI must run in the same HA resource group as NNMi. For information about this configuration, see Configuring the NNM iSPI for Performance as an Add-on iSPI on page 89.
- If the NNM iSPI for Performance is on a dedicated server, the iSPI can run in a separate HA resource group in the NNMi HA cluster, or the iSPI can run outside of HA. In either case, the iSPI connects to the virtual hostname of the NNMi HA resource group. For information about the HA scenario, see Configuring the NNM iSPI for Performance on a Dedicated Server for HA on page 91.

As part of NNMi HA configuration, the NNM iSPI for Performance enablement script must be run on each NNMi management server in the cluster.

The NNM iSPI for Performance delivers the `PerfSPIHA` Perl module, which contains custom implementations of the HA configuration commands. When running the HA configuration commands, use the value `PerfSPIHA` in place of *`<iSPI_PM_Name>`* in the instructions.

# Using the NNM iSPI for Performance

The NNM iSPI for Performance adds several reporting actions to the NNMi console. Each of these actions starts the NNM iSPI for Performance user interface at the designated level.

For information about the NNM iSPI for Performance reports, see the NNM iSPI for Performance help.

# Backup Provisions for the NNM iSPI for Performance

The NNMi backup scripts only back up the performance monitoring configuration. NNMi does *not* provide a mechanism for backing up any performance data.

To back up the performance data, you must use select and implement a separate backup tool on the NNM iSPI for Performance system. Add the deploy folder and all of its subfolders to this separate backup process.

The NNM iSPI for Performance stores data in the following folder:

- *Windows*: `%NnmInstallDir%\NNMPerformanceSPI\deploy\`
- *UNIX*: `$NnmInstallDir/NNMPerformanceSPI/deploy/`

# Resetting the NNM iSPI for Performance

If you want to completely reset your NNMi configuration and database so that you can start again with a clean slate, you must also reset the NNM iSPI for Performance configuration to keep the topology identifiers in agreement between the two products.

After you reset the NNMi database, run the NNM iSPI for Performance reset utility as described in *Resetting the Performance SPI* in the NNM iSPI for Performance help.

# Moving the NNM iSPI for Performance to a Different Computer

You can include the NNM iSPI for Performance in your NNMi test environment. The `nnmconfigexport.ovpl` command exports the NNMi performance monitoring configuration so that you can replicate the test configuration on the production NNMi management server.

The NNM iSPI for Performance reports are based on unique identifiers for the objects in the NNMi topology. Because these identifiers will be different on the new NNMi management server, you cannot port existing reports from the test system to the production system.

For information on the scenario for moving NNMi from a test system to the production environment, see Changing the NNMi Management Server on page 131.

# Disabling the NNM iSPI for Performance

For information about uninstalling the NNM iSPI for Performance, see *Uninstalling the Performance SPI* in the NNM iSPI for Performance help.

# Troubleshooting the NNM iSPI for Performance

For information about uninstalling the NNM iSPI for Performance, see *Troubleshooting* in the NNM iSPI for Performance help.

# Additional Information

This section contains the following appendices:

- NNMi Environment Variables
- The Causal Engine and NNMi Incidents

# NNMi Environment Variables

HP Network Node Manager i Software provides many environment variables that are available for your use in navigating the file system and writing scripts.

This appendix contains the following topics:


- Environment Variables Used in This Document
- Other Available Environment Variables
- Windows Paths and Environment Variables from NNMi 8.00


## Environment Variables Used in This Document

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

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

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

➤ If you first installed NNMi 8.00, your system uses different values for these environment variables, as described in Windows Paths and Environment Variables from NNMi 8.00 on page 238.

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

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

# Other Available Environment Variables

NNMi administrators access some NNMi file locations regularly. NNMi provides a script that sets up many environment variables for navigating to commonly accessed locations.

If you first installed NNMi 8.00, see Windows Paths and Environment Variables from NNMi 8.00 on page 238.

To set up the extended list of NNMi environment variables, use a command similar to the following examples:

- Windows: `"C:\Program Files\HP\HP BTO Software\bin\nnm.envvars.bat"`
- UNIX: `. /opt/OV/bin/nnm.envvars.sh`

After you run the command for your operating system, you can use the NNMi environment variables shown in Table 22 (Windows) or Table 23 (UNIX) to get to commonly used NNMi file locations.

**Table 22 Environment Variable Locations for the Windows Operating System**

| Variable | Windows (example) |
|---|---|
| %NNM_BIN% | C:\Program Files\HP\HP BTO Software\bin |
| %NNM_CONF% | C:\Documents and Settings\All Users\Application Data\HP\HP BTO Software\conf |
| %NNM_DATA% | C:\Documents and Settings\All Users\Application Data\HP\HP BTO Software |
| %NNM_DB% | C:\Documents and Settings\All Users\Application Data\HP\HP BTO Software\databases |
| %NNM_JAVA% | C:\Program Files\HP\HP BTO Software\nonOV\jdk\b\bin\java.exe |
| %NNM_JAVA_DIR% | C:\Program Files\HP\HP BTO Software\java |
| %NNM_JAVA_PATH_SEP% | ; |
| %NNM_JBOSS% | C:\Program Files\HP\HP BTO Software\nonOV\jboss\nms |
| %NNM_JBOSS_DEPLOY% | C:\Program Files\HP\HP BTO Software\nonOV\jboss\nms\server\nms\deploy |
| %NNM_JBOSS_LOG% | C:\Program Files\HP\HP BTO Software\nonOV\jboss\nms\server\nms\log |
| %NNM_JBOSS_ROOT% | C:\Program Files\HP\HP BTO Software\nonOV\jboss\nms |
| %NNM_JBOSS_SERVERCONF% | C:\Program Files\HP\HP BTO Software\nonOV\jboss\nms\server\nms |
| %NNM_JRE% | C:\Program Files\HP\HP BTO Software\nonOV\jdk\ |
| %NNM_LOG% | C:\Documents and Settings\All Users\Application Data\HP\HP BTO Software\log |
| %NNM_PRIV_LOG% | C:\Documents and Settings\All Users\Application Data\HP\HP BTO Software\log |

**Table 22 Environment Variable Locations for the Windows Operating System**

| Variable | Windows (example) |
|---|---|
| %NNM_SHARE_LOG% | C:\Documents and Settings\All Users\Application Data\HP\HP BTO Software\log |
| %NNM_SNMP_MIBS% | C:\Documents and Settings\All Users\Application Data\HP\HP BTO Software\share\snmp_mibs |
| %NNM_SUPPORT% | C:\Program Files\HP\HP BTO Software\support |
| %NNM_TMP% | C:\Documents and Settings\All Users\Application Data\HP\HP BTO Software\tmp |
| %NNM_WWW% | C:\Program Files\HP\HP BTO Software\www |

**Table 23 Environment Variable Locations for UNIX Operating Systems**

| Variable | HP-UX |
|---|---|
| $NNM_BIN | /opt/OV/bin |
| $NNM_CONF | /var/opt/OV/conf |
| $NNM_DATA | /var/opt/OV |
| $NNM_DB | /var/opt/OV/databases |
| $NNM_JAVA | /opt/OV/nonOV/jdk/b/bin/java |
| $NNM_JAVA_DIR | /opt/OV/java |
| $NNM_JAVA_PATH_SEP | : |
| $NNM_JBOSS | /opt/OV/nonOV/jboss/nms |
| $NNM_JBOSS_DEPLOY | /opt/OV/nonOV/jboss/nms/server/nms/deploy |
| $NNM_JBOSS_LOG | /opt/OV/nonOV/jboss/nms/server/nms/log |
| $NNM_JBOSS_ROOT | /opt/OV/nonOV/jboss/nms |
| $NNM_JBOSS_SERVERCONF | /opt/OV/nonOV/jboss/nms/server/nms |
| $NNM_JRE | /opt/OV/nonOV/jdk/b |
| $NNM_LOG | /var/opt/OV/log |
| $NNM_PRIV_LOG | /var/opt/OV/log |
| $NNM_SHARE_LOG | /var/opt/OV/log |
| $NNM_SNMP_MIBS | /var/opt/OV/share/snmp_mibs |
| $NNM_SUPPORT | /opt/OV/support |
| $NNM_TMP | /var/opt/OV/tmp |
| $NNM_WWW | /opt/OV/www |

# Windows Paths and Environment Variables from NNMi 8.00

On the Windows operating system, the default paths to NNMi files are different for versions 8.00 and 8.01 (or higher). When you upgrade to a newer version of NNMi on top of NNMi 8.00, the newer version of NNMi continues to use the paths that were defined for NNMi 8.00. In this case, the locations of the environment variables created during NNMi installation are as follows:

- `%NnmInstallDir%`: `<drive>\Program Files (x86)\HP OpenView`
- `%NnmDataDir%`: `<drive>\Program Files (x86)\HP OpenView\data`

To set up the extended list of NNMi environment variables, use a command similar to the following example:

```
"C:\Program Files(x86)\HP OpenView\bin\nnm.envvars.bat"
```

After you run the command, you can use the NNMi environment variables shown in Table 24 to get to commonly used NNMi file locations.

For the upgrade scenario *only*, the information in Table 24 supersedes that in Table 22 on page 236.

**Table 24 NNMi 8.00 Environment Variable Locations for the Windows Operating System**

| Variable | Windows (example) |
|---|---|
| %NNM_BIN% | C:\Program Files (x86)\HP OpenView\bin |
| %NNM_CONF% | C:\Program Files (x86)\HP OpenView\data\conf |
| %NNM_DATA% | C:\Program Files (x86)\HP OpenView\data |
| %NNM_DB% | C:\Program Files (x86)\HP OpenView\data\databases |
| %NNM_JAVA% | C:\Program Files (x86)\HP OpenView\nonOV\jdk\b\bin\java.exe |
| %NNM_JAVA_DIR% | C:\Program Files (x86)\HP OpenView\java |
| %NNM_JAVA_PATH_SEP% | ; |
| %NNM_JBOSS% | C:\Program Files (x86)\HP OpenView\nonOV\jboss\nms |
| %NNM_JBOSS_DEPLOY% | C:\Program Files (x86)\HP OpenView\nonOV\jboss\nms\server\nms\deploy |
| %NNM_JBOSS_LOG% | C:\Program Files (x86)\HP OpenView\nonOV\jboss\nms\server\nms\log |
| %NNM_JBOSS_ROOT% | C:\Program Files (x86)\HP OpenView\nonOV\jboss\nms |
| %NNM_JBOSS_SERVERCONF% | C:\Program Files (x86)\HP OpenView\nonOV\jboss\nms\server\nms |
| %NNM_JRE% | C:\Program Files (x86)\HP OpenView\nonOV\jdk\b |
| %NNM_LOG% | C:\Program Files (x86)\HP OpenView\data\log |
| %NNM_PRIV_LOG% | C:\Program Files (x86)\HP OpenView\data\log |
| %NNM_SHARE_LOG% | C:\Program Files (x86)\HP OpenView\data\log |

**Table 24 NNMi 8.00 Environment Variable Locations for the Windows Operating System**

| Variable | Windows (example) |
|---|---|
| %NNM_SNMP_MIBS% | C:\Program Files (x86)\HP OpenView\data\share\snmp_mibs |
| %NNM_SUPPORT% | C:\Program Files (x86)\HP OpenView\support |
| %NNM_TMP% | C:\Program Files (x86)\HP OpenView\data\tmp |
| %NNM_WWW% | C:\Program Files (x86)\HP OpenView\www |

# The Causal Engine and NNMi Incidents

Communications and data networks have grown significantly in size and complexity, and so have the number of faults that occur. A single failure can trigger many alarms, and the bottleneck for the network operator has become distinguishing the real problem from all the anecdotal alarms. Traditional event correlation systems have been able to reduce alarms, but these systems tend to fall short in terms of identifying the root cause in an automated way.

The HP Network Node Manager i Software Causal Engine technology applies **root cause analysis (RCA)** to network symptoms, using a causality-based approach.

This appendix contains the following topics:


- Causality Analysis - High Level Discussion
- Causal Engine Concepts
- Concept of Status
- What is an Episode?
- What Does NNMi Analyze?
- What Are the Failure Scenarios?
- Network Configuration Changes
- NNMi Management Configuration Changes


## Causality Analysis - High Level Discussion

Causal Engine technology provides the following high-level features:

- Uses the `NmsApa` jboss service to analyze your network
- Model based approach to RCA
  - Models the behavioral relationship between managed objects
  - Uses an object model in addition to event causality to drive analysis
  - Determines root cause and impact
  - Based on the MINCAUSE algorithm
  - Able to deal with ambiguity and partial symptoms
- Dynamic
  - Actively solicits symptoms during analysis
  - Reacts dynamically to topology change
- Extensible
  - Employs a hierarchy of modules (import/export)
  - Provides an end-to-end diagnosis of network faults
  - Provides the ability to add rule sets in future offerings

# Causal Engine Concepts

Causal Engine technology uses the following sequential approach:

1. Formally define the root-cause problems and symptoms.
2. Perform analysis by relating symptoms to root-cause problems using the model. Symptoms come from two sources:
   - StatePoller, where the symptoms are state changes.
   - Events, where the symptoms are traps.
3. Generate conclusions that relate to the root cause.

   Causal Engine conclusions contain artifacts that are associated with the model. Artifacts include the following details:
   - Incident generation
   - Incident correlation
   - Incident suppression
   - Incident cancellation
   - Status on relevant objects

# Concept of Status

In addition to incident manipulation, the `NmsApa` service sets status on relevant objects. Status is used to indicate the overall health of an object and is calculated as a result of outstanding conclusions. Every conclusion has a severity associated with it; the status reported is the most severe of all outstanding conclusions. In addition, conclusions inform the user of the underlying *cause* (or reason) for an object's status.

The `NmsApa` service manages the following objects:

- SNMP Agent
- IPv4Address
- Interface
- Connection
- Node
- Node Group

The `NmsApa` service uses the following status categories in decreasing order of severity:

- Unknown
- Disabled
- Critical
- Minor
- Warning

- Normal
- No Status

## What is an Episode?

The goal of the `NmsApa` service is to present a single incident that the operator or network engineer can work with. To do this the `NmsApa` service uses the concept of an **episode**. An episode exists for a specific duration, during which secondary failures are either correlated or suppressed based on configuration. For example:

- The `AddressNotResponding` incident is suppressed by the `InterfaceDown` incident, according to the following scenario:
  - When an IPv4 address stops responding to ICMP, an episode is started, which exists for a duration of 60 seconds.
  - Within that duration, if the interface associated with that IPv4 address goes down, the `NmsApa` service concludes that the *interface down* condition caused the IPv4 address to stop responding.
  - Therefore, the `AddressNotResponding` incident is not generated. Only the `InterfaceDown` incident is generated.
  - In order to ensure that the `InterfaceDown` incident is detected within the duration, the `NmsApa` service issues a *named poll* for that interface. This enables the network engineer to fix the root cause of the problem which, in this case, is the interface.
  - If the interface does not go down during the episode, the `NmsApa` service generates an `AddressNotResponding` incident. If the interface goes down after the episode, the `InterfaceDown` incident is generated. In this case, the network engineer has to treat the two problems separately.
- The `NodeDown` incident correlates the `InterfaceDown` incident from one-hop neighbor interfaces, according to the following scenario:
  - When an interface goes down, a `NodeDown` episode is started for the neighboring node, which exists for a duration of 900 seconds.
  - Within that duration, if the node goes down, the `InterfaceDown` incident is correlated beneath the `NodeDown` incident.
  - The `InterfaceDown` incidents from all one-hop neighbors are correlated beneath the `NodeDown` incident. You can review the `InterfaceDown` incidents as supporting evidence for the `NodeDown` incident.

## What Does NNMi Analyze?

NNMi uses SNMP protocol to retrieve information from a managed node, using an SNMP agent (a process running on the managed node, which provides management functions). The SNMP agent is responsible for managing interfaces and ports on the managed node; it can be associated with one or more nodes.

The following list shows the possible NNMi status categories associated with an SNMP agent:

- `Unknown` - Not applicable.
- `Disabled` - Not applicable.
- `Critical` - SNMP Agent doesn't respond to SNMP queries.
- `Minor` - Not applicable.
- `Warning` - Not applicable.
- `Normal` - SNMP Agent responds to SNMP queries.
- `No Status` - SNMP Agent is not polled.

An **IPv4 address** is a routable address that responds to ICMP. IPv4 addresses are typically associated with nodes. NNMi reports the status of a node as follows:

- `Unknown` - Not applicable.
- `Disabled` - The interface associated with this IPv4 address is administratively down or disabled.
- `Critical` - IPv4 address doesn't respond to ICMP queries (ping the device).
- `Minor` - Not applicable.
- `Warning` - Not applicable.
- `Normal` - IPv4 address responds to ICMP queries.
- `No Status` - IPv4 address is not polled.

An **interface** is a physical port that can be used to connect a node to the network. NNMi reports the status of an interface as follows:

- `Unknown` - The SNMP Agent associated with the interface doesn't respond to SNMP queries. Unknown indicates that the `NmsApa` service cannot determine the health because `ifAdminStatus` and `ifOperStatus` cannot be measured.
- `Disabled` - Interface is administratively down (`ifAdminStatus` = down).
- `Critical` - Interface is operationally down (`ifOperStatus` = down).
- `Minor` - Not applicable.
- `Warning` - Not applicable.
- `Normal` - Interface is operationally up (`ifOperStatus` = up).
- `No Status` - Interface is not polled.

A **node** is a device that NNMi finds as a result of the spiral discovery process. A node can contain interfaces, boards, and ports. You can separate nodes into two categories:

1. Network nodes, active devices such as switches, routers, bridges, and hubs
2. End nodes, such as UNIX or Windows servers

NNMi typically manages network nodes, reporting the node status as follows:

- `Unknown` - The SNMP Agent associated with the node doesn't respond to SNMP queries and polled IPv4 addresses do not respond to ICMP queries. This indicates that NNMi is unable to manage the node.
- `Disabled` - Not applicable.

- `Critical` - Any one of the following:
  - The node is down as determined by neighbor analysis.
  - The node is marked as important and it is *unmanageable* (NNMi cannot access the node from the NNMi server).
  - The node is an island (it has no neighbors), and therefore, it is unmanageable.
  - The `NmsApa` service cannot determine if the node is down or if the incoming connection is down.
- `Minor` - Any one of the following:
  - The SNMP Agent associated with the node doesn't respond to SNMP queries.
  - One or more interfaces in the node are down.
  - One or more IPv4 addresses on the node are not responding to ICMP.
- `Warning` - Not applicable.
- `Normal` - The SNMP Agent, polled interfaces, and polled IPv4 addresses of the node are up.
- `No Status` - The SNMP Agent, all interfaces, and all IPv4 addresses of the node are not polled.

**Connections** are Layer 2 physical connections and Layer 3 network connections. NNMi discovers connection information by reading forwarding database (FDB) tables from other network devices and by using devices that support discovery protocols such as CDP and EDP. NNMi reports the status of a connection as follows:

- `Unknown` - All endpoints of the connection have unknown status.
- `Disabled` - Any one endpoint of the connection is disabled.
- `Critical` - All endpoints are operationally down.
- `Minor` - Any one endpoint is down.
- `Warning` - Endpoints have unknown and non-critical status.
- `Normal` - All endpoints are operationally up.
- `No Status` - Any one endpoint is not polled.

A **node group** is a logical collection of nodes used to separate the polling configuration. An administrator creates node-type groupings. For example, some nodes, such as routers, are critical to your business, so you may want to poll these routers more frequently. To do this, you define a node group containing the critical routers and configure them for a shorter polling cycle.

NNMi reports the status of a node group as follows:

- `Unknown` - All nodes in the group have unknown status.
- `Disabled` - Not applicable.
- `Critical` - All nodes in the group have critical status.
- `Minor` - One or more nodes in the group have critical status.
- `Warning` - Nodes have status of unknown and non-critical status.
- `Normal` - All nodes in the group have normal status.
- `No Status` - All nodes in the group have no status.

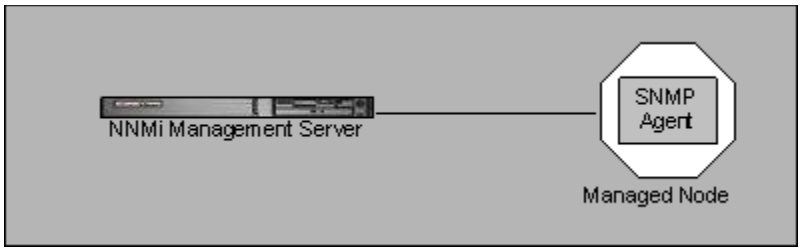
# What Are the Failure Scenarios?

The following scenarios show examples of problems in the network and the work that the Causal Engine does to diagnose these problems. These scenarios show you examples of the incidents shown in Table 25, as well as other examples.

**Table 25 Incident Definitions**

| Incident Name | Description |
|---|---|
| AddressNotResponding | An IPv4 address is not responding to ICMP, possibly for the following reasons:<br>1 The node is down.<br>2 A device, such as a router, has been mis-configured so some IPv4 addresses cannot be reached. |
| InterfaceDown | The interface is not responding to polls. |
| ConnectionDown | Both (or all) ends of a connection are down. |
| ConnectionPartiallyUnresponsive | This incident indicates that a connection is partially unresponsive. |
| NodeDown | This incident indicates that the `NmsApa` service has determined the node is down based on the following analysis:<br>• 100% of the IPv4 addresses assigned to this node are unreachable.<br>• The SNMP agent installed on this machine is not responding. At least two of the neighboring devices can be reached and are reporting problems with connectivity to this node. |
| NodeOrConnectionDown | This incident indicates that a node is not responding to an ICMP or SNMP query. It also indicates that only one neighbor is down so that the `NmsApa` service cannot determine whether the node or the connection is down. |

## SNMP Agent Not Responding to SNMP Queries




### Legend

| | |
|---|---|
| Managed Node | Network device, such as an Ethernet switch |
| SNMP Agent | With new community string for managed node |
| MS communication configuration settings | Not updated with new community strings |

**Scenario**: The SNMP agent is not responding. For instance, the community string for this SNMP agent has been changed, NNMi's communication configuration settings have not yet been updated, but the node is operational (its IPv4 addresses can be pinged).

**Root Cause**: The SNMP Agent is not responding.

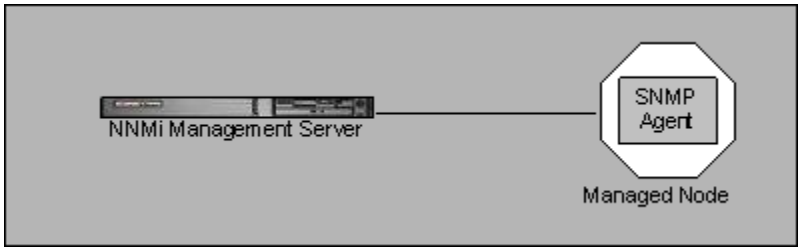
**Incident**: None generated.

**Status**: SNMP Agent is in critical status.

**Conclusion**: `SNMPAgentNotResponding`

**Effect**: Node status is minor; conclusion on the node is `UnresponsiveAgentInNode`. All polled interfaces have unknown status because they cannot be managed by NNMi. The conclusion on each interface is `InterfaceUnmanageable`.

## SNMP Agent Responding to SNMP Queries




Legend

| | |
|---|---|
| Managed Node | Network device, such as an Ethernet switch |
| SNMP Agent | With new community string for managed node |
| MS communication configuration settings | Updated with new community strings |

**Scenario**: See page 247 because this scenario continues the *SNMP Agent Not Responding to SNMP Queries* scenario. Suppose an NNMi administrator updates the communication configuration settings to include the new community string. The SNMP agent for the managed node starts responding to SNMP queries.

**Root Cause**: SNMP Agent is responding.

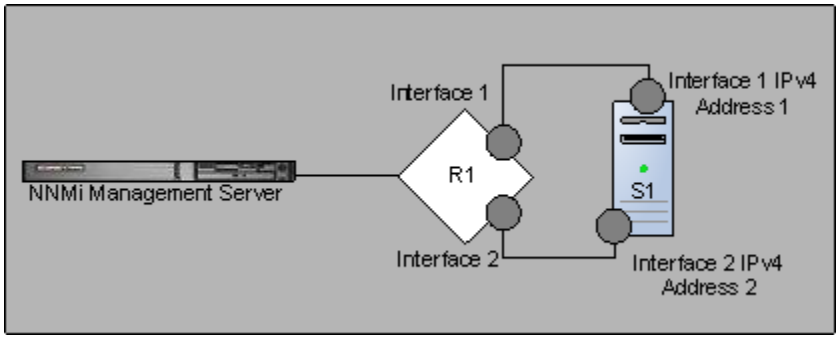
**Incident**: None generated.

**Status**: SNMP Agent is in normal status.

**Conclusion**: `SNMPAgentResponding`

**Effect**: Node status is normal; conclusion on the node is `ResponsiveAgentInNode`. `InterfaceUnmanagable` is cleared from all polled interfaces and the interfaces revert to their previous status.

## IPv4Address Not Responding to ICMP




### Legend

| | |
|---|---|
| R1 | Router 1 |
| Route | Changed from interface 1 to interface 2 on the router |
| S1 | Server 1 |
| Managed Node S1 | A multi-homed server |
| S1 Interface 1 | Associated with IPv4 address 1 |
| S1 Interface 2 | Associated with IPv4 address 2 |

**Scenario**: IPv4 address 1 on S1 is not responding. For instance, suppose the route on Router 1 (R1) changes from interface 1 to interface 2, so that packets destined for S1 interface 1 are now routed out of R1 interface 2. The associated interface is operational, and the node can be reached because you can ping some IPv4 addresses. The SNMP agent is up.

**Root Cause**: IPv4 address is not responding.

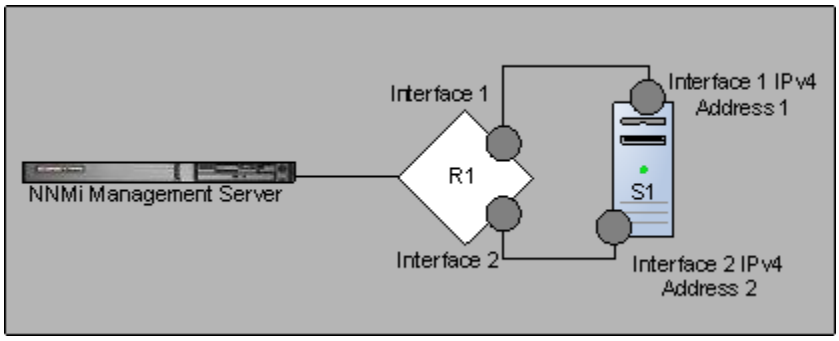
**Incident**: `AddressNotResponding` incident generated.

**Status**: IPv4 address is in critical status.

**Conclusion**: `AddressNotResponding`

**Effect**: Node status is minor; conclusion on the node is `SomeUnresponsiveAddressesInNode`.

## IPv4Address Responding to ICMP




### Legend

| | |
|---|---|
| R1 | Router 1 |
| Route | Changed from interface 2 to interface 1 on the router |
| S1 | Server 1 |
| Managed Node S1 | A multi-homed server |
| S1 Interface 1 | Associated with IPv4 address 1 |
| S1 Interface 2 | Associated with IPv4 address 2 |

**Scenario**: See page 249 because this scenario continues the *IPv4Address Not Responding to ICMP* scenario. Suppose that the IPv4 address is now responding, the associated interface is operational, and the node can be reached (for example, you can ping some IPv4 addresses, and/or the SNMP agent is up).

**Root Cause**: IPv4 address is responding.

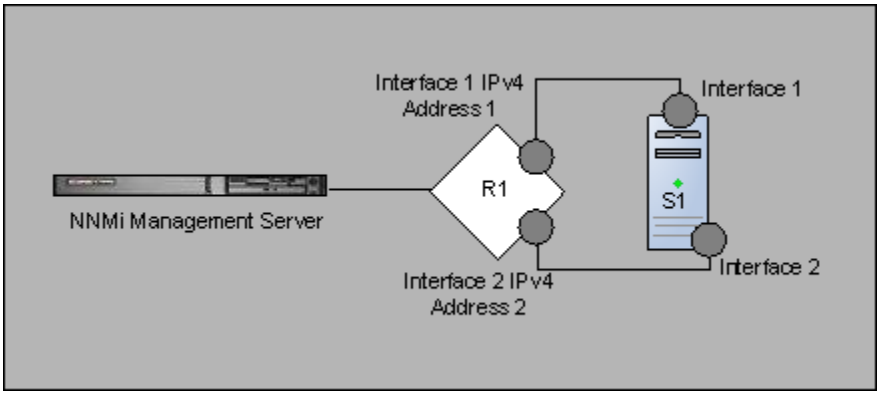
**Incident**: None generated; the `AddressNotResponding` incident closed.

**Status**: IPv4 address is in normal status.

**Conclusion**: `AddressResponding`

**Effect**: Node status is normal; conclusion on the node is `ResponsiveAddressesInNode`.

## Interface is Operationally Down




### Legend

| | |
|---|---|
| R1 | Router 1 |
| Interface 1 for R1 | Configured to be administratively up but is operationally down |
| Interface 1 for R1 | Configured for IPv4 address 1 |
| Interface 2 for Router 1 | Configured for IPv4 address 2 |
| S1 | Server 1 |

**Scenario**: R1 interface 1 is operationally down (`ifOperStatus` = down) and administratively up (`ifAdminStatus` = up). R1 sends a `linkDown` trap. R1 can be reached because you can ping some IPv4 addresses, such as IPv4 address 2. The SNMP agent is up. IPv4 address 1 is associated with interface 1; it has stopped responding to ICMP.

**Root Cause**: Interface is down.

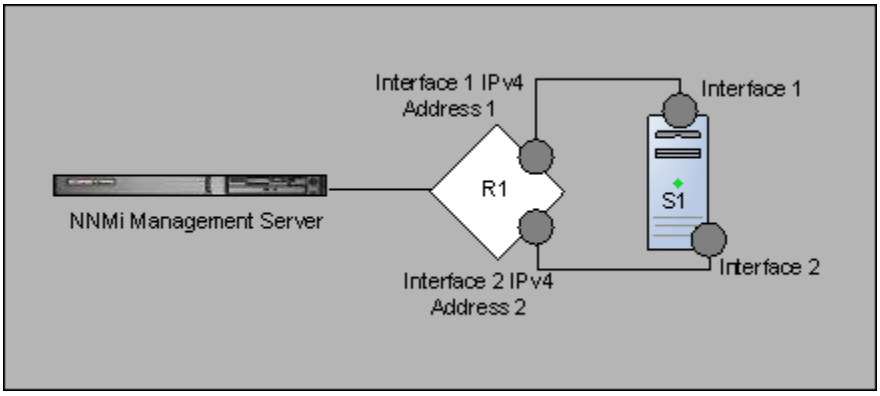
**Incident**: `InterfaceDown` incident generated; the `LinkDown` incident is correlated beneath the `InterfaceDown` incident.

**Status**: Interface is in critical status.

**Conclusion**: `InterfaceDown`

**Effect**: Node status is minor; conclusion on the node is `InterfacesDownInNode`. No `AddressNotResponding` incident is associated with the IPv4 address.

## Interface is Operationally Up




### Legend

| | |
|---|---|
| R1 | Router 1 |
| Interface 1 for R1 | Configured to be administratively up and operationally up |
| Interface 1 for R1 | Configured for IPv4 address 1 |
| Interface 2 for Router 1 | Configured for IPv4 address 2 |
| S1 | Server 1 |

**Scenario**: See page 251 because this scenario continues the *Interface is Operationally Down* scenario. Suppose that R1 interface 1 is now operationally up (`ifOperStatus` = up). The node is reachable; you can ping all of its IPv4 addresses. The SNMP agent is up.

**Root Cause**: Interface is up.

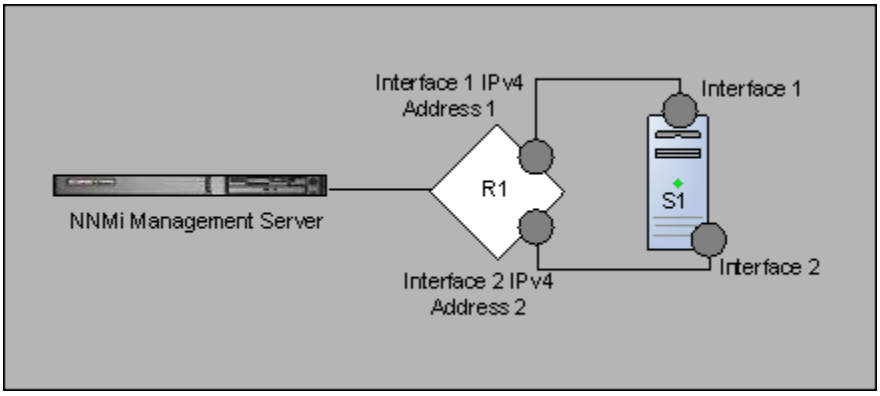
**Incident**: None generated; the `InterfaceDown` incident is closed.

**Status**: Interface is in normal status.

**Conclusion**: `InterfaceUp`

**Effect**: Node status is normal; conclusion on the node is `InterfacesUpInNode`.

## Interface is Administratively Down




### Legend

| | |
|---|---|
| R1 | Router 1 |
| Interface 1 for R1 | Configured to be administratively up and operationally down |
| Interface 1 for R1 | Configured for IPv4 address 1 |

**Scenario**: R1 interface 1 is administratively down (`ifAdminStatus` = down), but the node can be reached. For instance, you can ping interface 2 and the SNMP agent is up. Disabling R1 interface 1 brings that interface operationally down. The IPv4 address associated with this interface, IPv4 address 1, stops responding to ICMP.

**Root Cause**: R1 interface 1 is disabled.

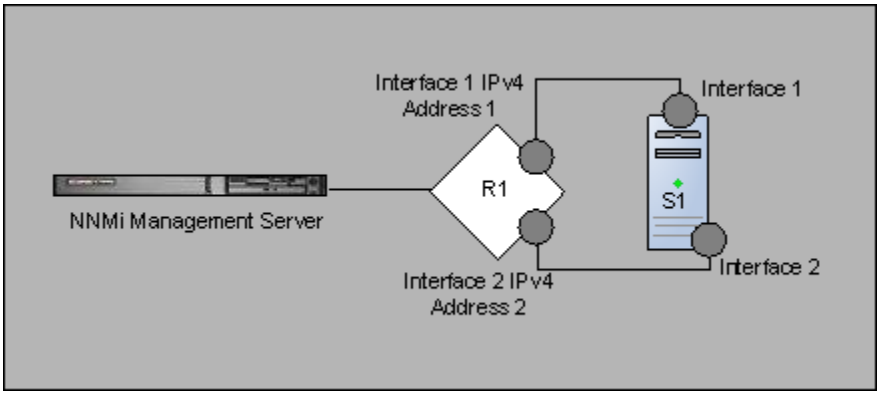
**Incident**: None generated.

**Status**: Interface is in disabled status.

**Conclusion**: `InterfaceDisabled`

**Effect**: The IPv4 address associated with R1 interface 1 has a status of `disabled`; the conclusion on the IPv4 address is `AddressDisabled`.

## Interface is Administratively Up




### Legend

| | |
|---|---|
| MS | NNMi Management Server |
| R1 | Router 1 |
| Interface 1 for R1 | Configured to be administratively up and operationally up |
| Interface 1 for R1 | Configured for IPv4 address 1 |

**Scenario**: See page 253 because this scenario continues the *Connection is Administratively Down* scenario. Suppose R1 interface 1 is now administratively up (`ifAdminStatus` = up), and you can reach the node by pinging some IPv4 addresses of that interface. The SNMP agent is up. Enabling R1 interface 1 brings it operationally up. The IPv4 address associated with this interface starts responding to ICMP.

**Root Cause**: Interface is enabled.

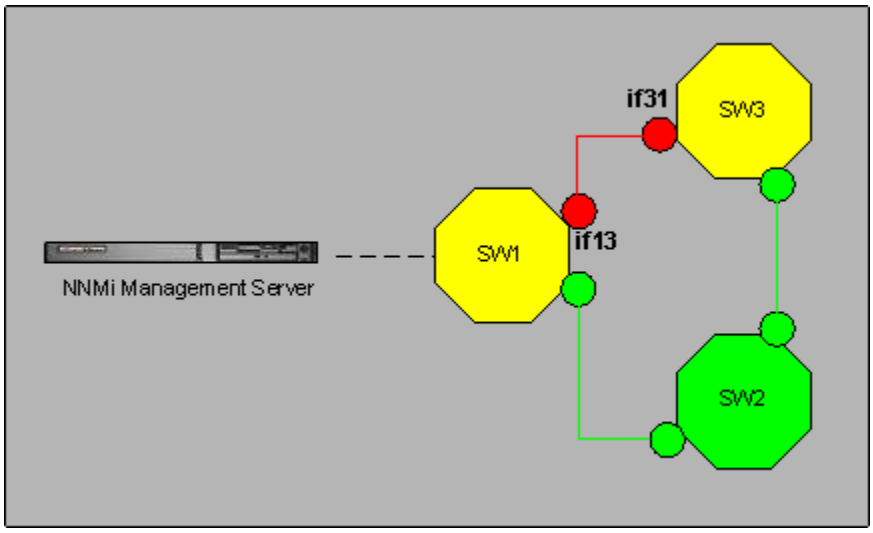
**Incident**: None generated.

**Status**: Interface is in normal status.

**Conclusion**: `InterfaceEnabled`

**Effect**: The IPv4 address associated with R1 interface 1 has a status of `enabled`; the conclusion on the IPv4 address is `AddressEnabled`.

## Connection is Operationally Down




### Legend

| | |
|---|---|
| SW1 | Switch 1 |
| SW2 | Switch 2 |
| SW3 | Switch 3 |
| IF31 | Interface on Switch 3 connecting to Switch 1 |
| IF13 | Interface on Switch 1 connecting to Switch 3 |

**Scenario**: The connection between the interface on Switch 3 connecting to Switch 1 (IF13) and the interface on Switch 1 connecting to Switch 3 (IF31) is down. Traffic flows from the Management Server through Switch 1 (SW1) **and** Switch 2 (SW2). Both IF13 and IF31 are marked `down`.

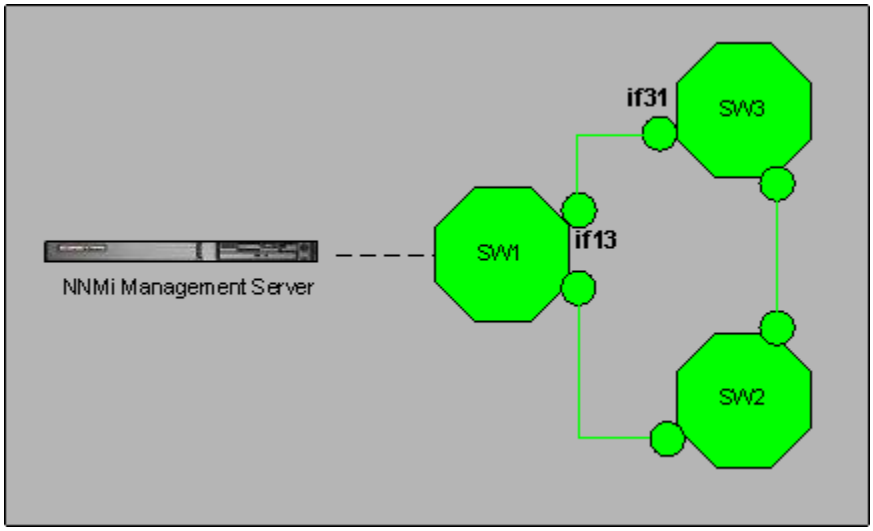
**Root Cause**: The connection between IF13 and IF31 is down.

**Incident**: `ConnectionDown` incident generated; the `InterfaceDown` incidents from IF13 and IF31 are correlated beneath `ConnectionDown`.

**Status**: Connection is in critical status.

**Conclusion**: `ConnectionDown`

## Connection is Operationally Up




**Scenario**: See page 255 because this scenario continues the *Connection is Operationally Down* scenario. Suppose that the connection between IF13 and IF31 is now up.

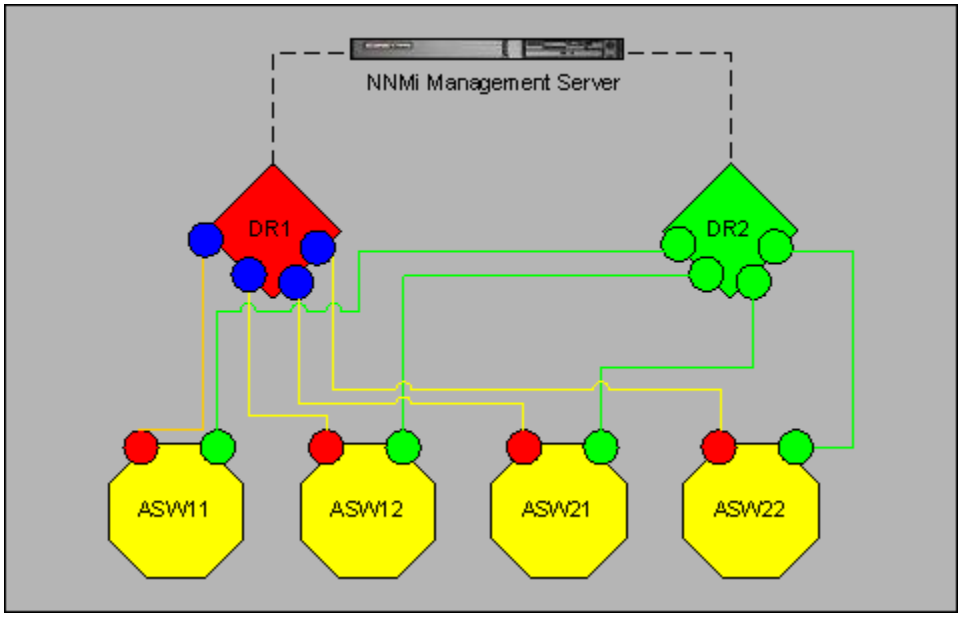
**Root Cause**: Connection between IF13 and IF31 is up.

**Incident:** None generated; the `ConnectionDown` incident is closed.

**Status**: Connection is in normal status.

**Conclusion**: `ConnectionUp`

## Directly Connected Node is Down




### Legend

| | |
|---|---|
| DR1 | Distribution Router 1 |
| DR2 | Distribution Router 2 |
| ASW11 | Access Switch 11 |
| ASW12 | Access Switch 12 |
| ASW21 | Access Switch 21 |
| ASW22 | Access Switch 22 |

**Scenario**: Suppose that access switches ASW11, ASW12, ASW21, and ASW22 are redundantly connected to the distribution routers as shown above. The distribution routers DR1 and DR2 are directly connected to one another. The distribution router DR1 goes down.

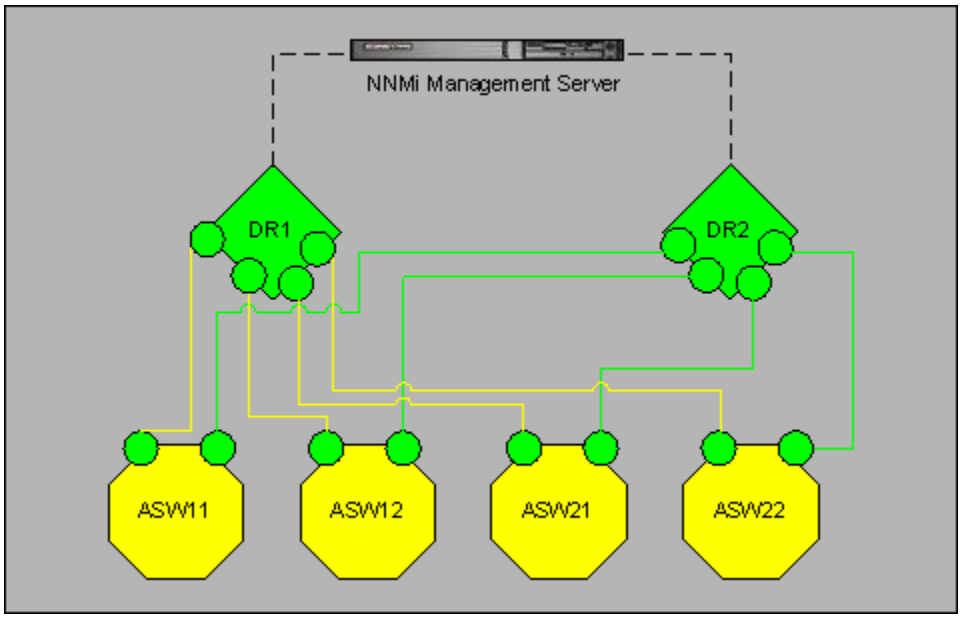
**Root Cause**: Node DR1 is down according to neighbor analysis.

**Incident**: `NodeDown` incident generated; the `InterfaceDown` incidents from one-hop neighbors are correlated beneath the `NodeDown` incident.

**Status**: Node is in critical status.

**Conclusion**: `NodeDown`

## Directly Connected Node is Up




Legend

| | |
|---|---|
| DR1 | Distribution Router 1 |
| DR2 | Distribution Router 2 |
| ASW11 | Access Switch 11 |
| ASW12 | Access Switch 12 |
| ASW21 | Access Switch 21 |
| ASW22 | Access Switch 22 |

**Scenario**: See page 257 because this scenario continues the *Directly Connected Node is Down* scenario. Suppose that the distribution router DR1 comes back up.

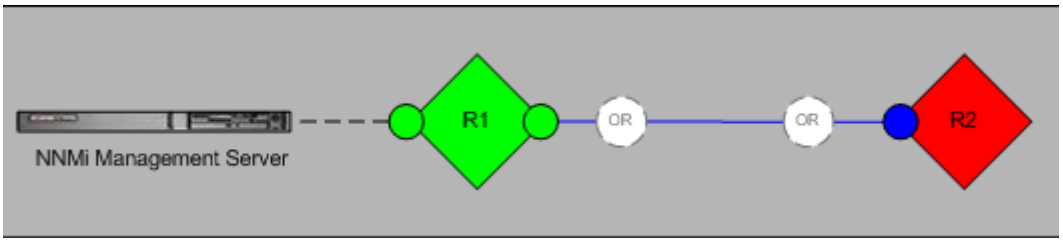
**Root Cause**: Node DR1 is up.

**Incident**: None generated; the `NodeDown` incident is closed by the `NodeUp` incident.

**Status**: Node is in normal status.

**Conclusion**: `NodeUp`

## Indirectly Connected Node is Down




Legend

| | |
|---|---|
| R1 | Router 1 |
| OR | optical repeaters (not discovered by NNMi) |
| R2 | Router 2 |

The diagram above is conceptual. It does not represent an actual NNMi topology map or workspace view.

**Scenario**: This scenario can occur with any indirect connection where NNMi cannot discover the intermediate devices. In this example, Routers R1 and R2 appear to be directly connected in NNMi topology maps, but in reality these two routers are indirectly connected via optical repeaters. (The optical repeaters do not respond to SNMP or ICMP queries, so they are not discovered by NNMi.)

Router R2 becomes unreachable, either because its connected interface is down or because the connection between the optical repeaters is down. The interface on Router R1 that indirectly connects it to Router R2 is still up, because its optical repeater is still up.

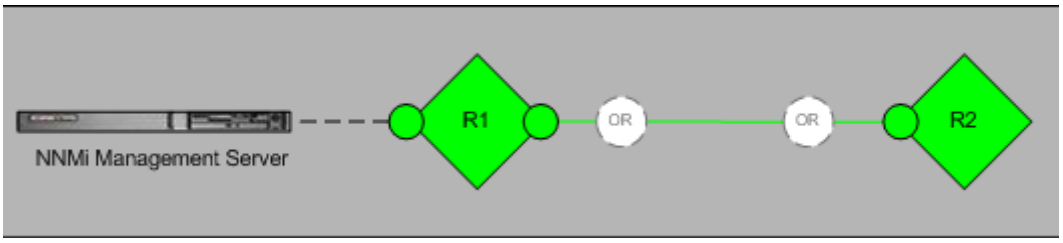
**Root Cause**: Router R2 is down according to neighbor analysis.

**Incident**: `NodeDown` incident generated.

**Status**: Node R2 is in critical status.

**Conclusion**: `NodeDown`

## Indirectly Connected Node is Up




Legend

| | |
|---|---|
| R1 | Router 1 |
| OR | optical repeaters (not discovered by NNMi) |
| R2 | Router 2 |

The diagram above is conceptual. It does not represent an actual NNMi topology map or workspace view.

**Scenario**: This scenario continues the previous *Indirectly Connected Node is Down* scenario. Suppose that the failed connection comes back up, and Router R2 becomes reachable.

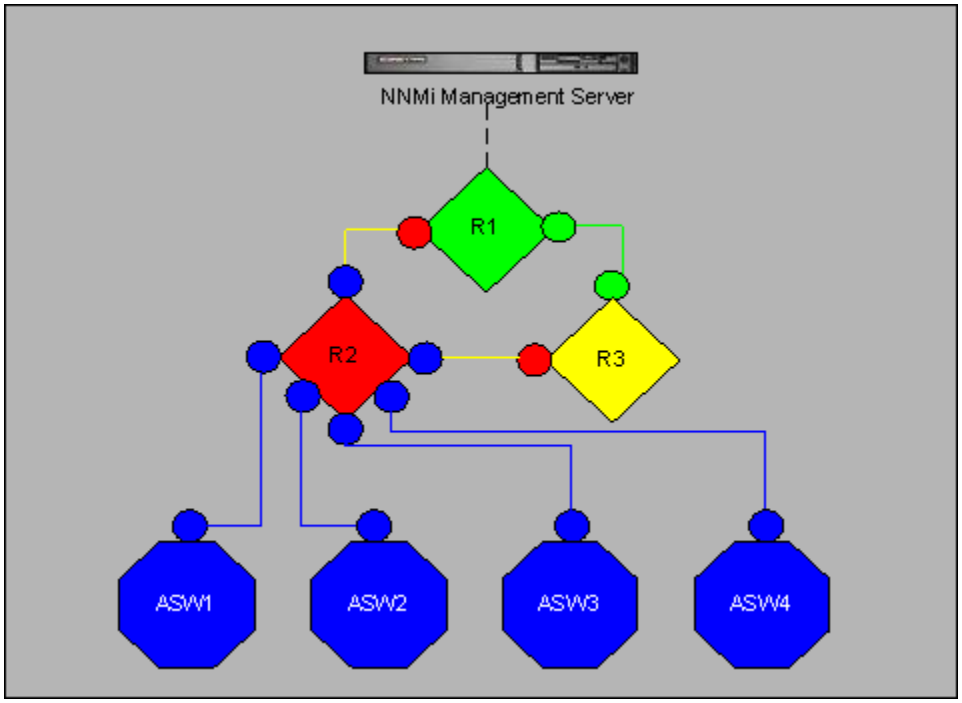
**Root Cause:** Connection between R1 and R2 is up.

**Incident:** None generated; the `NodeDown` incident is closed.

Status: Router R2's status is Normal. Connection status is Normal.

Conclusion: `NodeUp`

## Directly Connected Node is Down and Creates a Shadow




Legend

| | |
|---|---|
| R1 | Router 1 |
| R2 | Router 2 |
| R3 | Router 3 |
| ASW1 | Access Switch 1 |
| ASW2 | Access Switch 2 |
| ASW3 | Access Switch 3 |
| ASW4 | Access Switch 4 |

**Scenario**: The Router 2 (R2) goes down as shown above.

**Root Cause**: Node R2 is down according to NNMi's neighbor analysis.

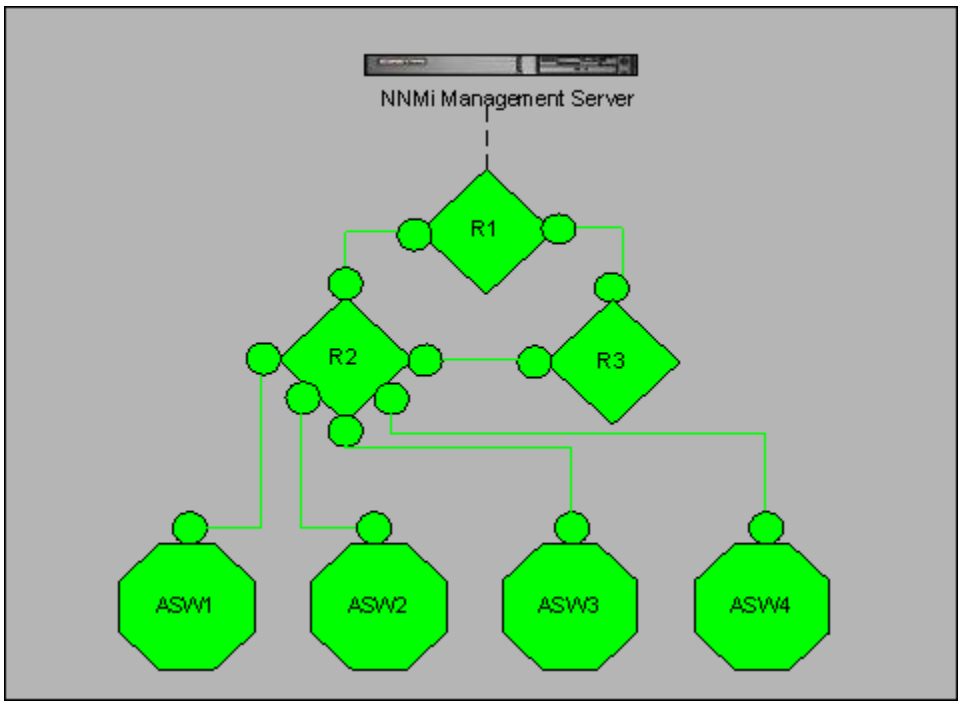
**Incident**: `NodeDown` incident generated; the `InterfaceDown` incidents from one-hop neighbors are correlated beneath the `NodeDown` incident.

**Status**: Node is in critical status.

**Conclusion**: `NodeDown`

**Effect**: All of the access switches are unreachable. The status of all nodes in the shadow is `unknown` and the conclusion on each of them is `NodeUnmanageable`.

## Directly Connected Node is Up, Clearing the Shadow




Legend

| | |
|---|---|
| R1 | Router 1 |
| R2 | Router 2 |
| R3 | Router 3 |
| ASW1 | Access Switch 1 |
| ASW2 | Access Switch 2 |
| ASW3 | Access Switch 3 |
| ASW4 | Access Switch 4 |

**Scenario**: See page 261 because this scenario continues the *Node is Down and Creates a Shadow* scenario. Suppose that R2 comes back up as shown above.

**Root Cause**: Node R2 is up.

**Incident**: None generated; the `NodeDown` incident is closed by the `NodeUp` incident.

**Status**: Node is in normal status.

**Conclusion**: `NodeUp`

**Effect**: All of the access switches are now reachable. The status of all nodes in the shadow is normal.

## Important Node is Unreachable

**Scenario**: A node is part of the `Important Nodes` node group and this node is unreachable.

You must add a node to the `Important Nodes` node group before the `NmsApa` service analyzes the node. If a node becomes unreachable before being added to the `Important Nodes` node group, the `NmsApa` service does not generate a `NodeDown` incident.

**Root Cause**: Node is down. The `NmsApa` service does not do neighbor analysis but concludes that the node is down simply because it was marked as important.

**Incident**: `NodeDown` incident generated; no correlated incidents.

**Status**: Node is in critical status.

**Conclusion**: `NodeDown`

## Important Node is Reachable

**Scenario**: See page 263 because this scenario continues the *Important Node is Unreachable* scenario. Suppose that the important node comes back up and is now reachable.

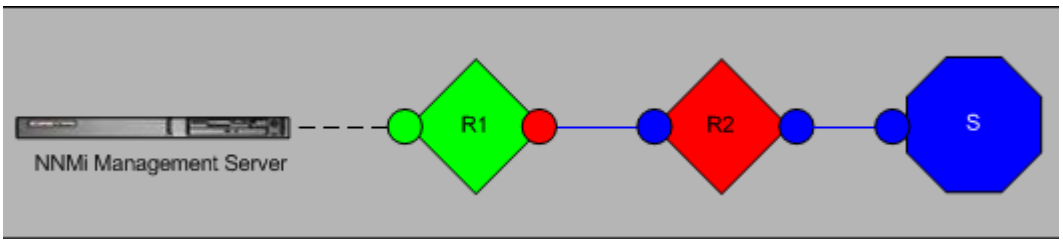
**Root Cause**: Node is up.

**Incident**: None generated; the `NodeDown` incident is closed by the `NodeUp` incident.

**Status**: Node is in normal status.

**Conclusion**: `NodeUp`

## Node or Connection is Down




Legend

| | |
|---|---|
| R1 | Router 1 |
| R2 | Router 2 |
| S | Access switch |

**Scenario**: There is no redundancy to Router 2 (R2). Either R2 is down or the connection between Router 1 (R1) and R2 is down.

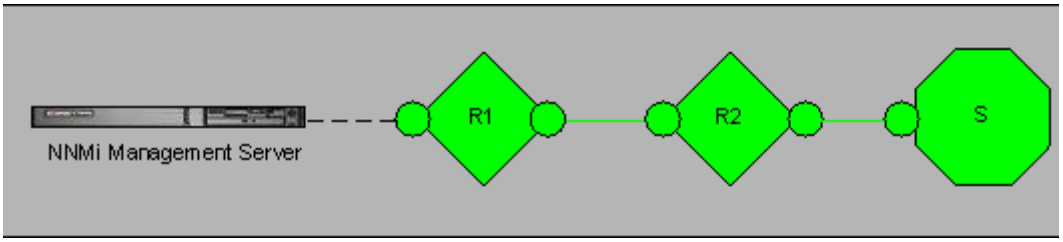
**Root Cause**: Node or connection is down.

**Incident**: `NodeOrConnectionDown` incident generated; the source node in this scenario is R2.

**Status**: Node is in critical status; connection is in warning status.

**Conclusion**: `NodeOrConnectionDown`

## Node or Connection is Up




Legend

| | |
|---|---|
| R1 | Router 1 |
| R2 | Router 2 |
| S | Access switch |

**Scenario**: See page 264 because this scenario continues the *Node or Connection is Down* scenario. Suppose that R2 is now up.

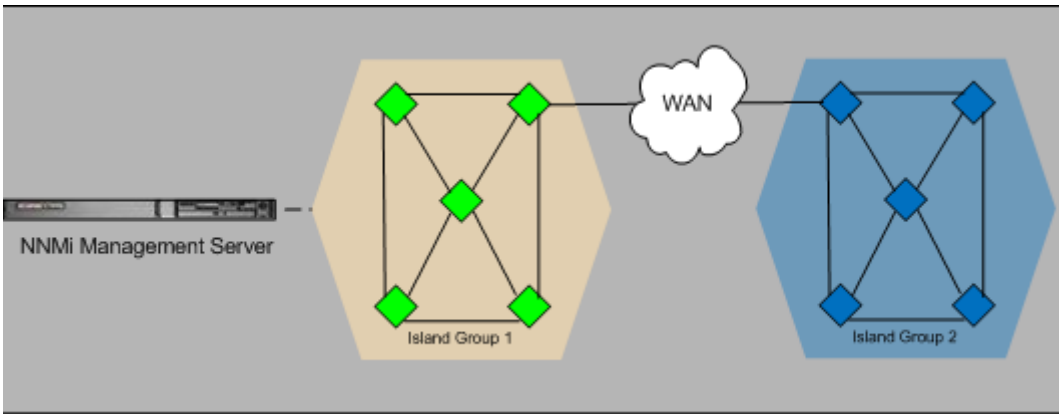
**Root Cause**: NodeUp

**Incident**: None generated; the `NodeOrConnectionDown` incident is closed.

**Status**: Node is in normal status; connection is in normal status.

**Conclusion**: `NodeUp`

## Island Group is Down




The diagram above is conceptual. It does not represent an actual NNMi topology map or workspace view.

**Scenario:** NNMi has partitioned your network into two Island Groups. The NNMi management server is connected to a node in Island Group 1. Island Group 2 has become unreachable due to problems in your service provider's WAN.

Island Groups contain highly-connected sets of nodes that are not connected or are only minimally connected to the rest of the network. For example, NNMi may identify multiple Island Groups for an enterprise network with geographically distributed sites connected by a WAN. Island Groups are created by NNMi and cannot be modified by the user. See the NNMi Console Help for more information on Island Groups.

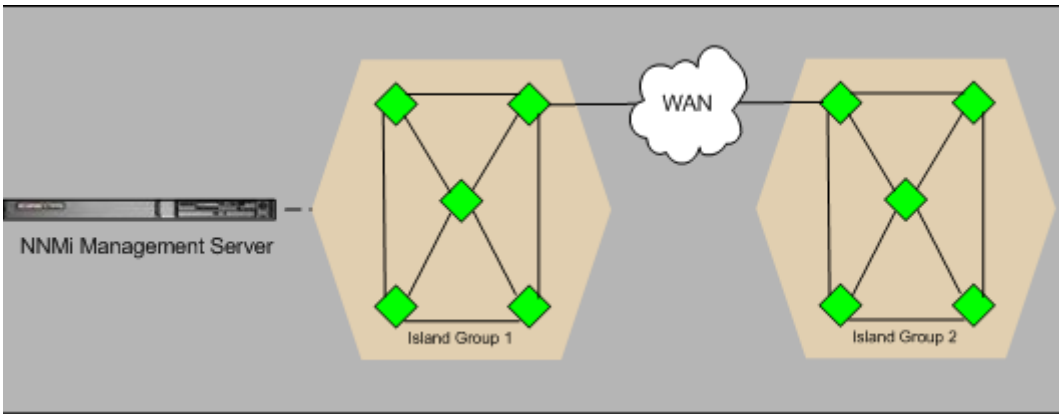
**Root Cause:** Island Group 2 is down according to neighbor analysis.

**Incident:** `IslandGroupDown` incident generated. NNMi uses a representative node from Island Group 2 as the source node for the incident.

**Status:** Island Group 2's status is set to Unknown. Objects in Island Group 2 have Unknown status. The connecting interface from Island Group 1 is Up, since its connection to the WAN is still up.

**Conclusion:** Not applicable for Island Groups

## Island Group is Up




The diagram above is conceptual. It does not represent an actual NNMi topology map or workspace view.

**Scenario:** This scenario continues the previous *Island Group is Down* scenario. Suppose that your service provider's WAN problems are fixed, and Island Group 2 becomes reachable.

**Root Cause:** WAN connection to Island Group 2 is back up.

**Incident:** None generated; the `IslandGroupDown` incident is closed.

**Status:** Island Group 2's status is set to Normal. Objects in Island Group 2 return to Normal status.

**Conclusion:** Not applicable for Island Groups

# Network Configuration Changes

During the span of a day, an NNMi operator may complete several configuration changes. The following scenarios illustrate some common network configuration changes and show how NNMi responds to these changes.

## Node Updated

Suppose that a network operator modifies a node: for example, by swapping a failed interface board with a working replacement. When NNMi notices this change, the discovery process sends a notification to the `NmsApa` service. The `NmsApa` service uses this notification to complete the following tasks:

- Recalculate the status of the node.
- Close all registered incidents for the deleted IPv4 addresses and interfaces on the node.

## Interface Moves to and from Connections

Suppose that a network operator changes the way network devices are connected. When an interface joins a connection or leaves one connection to join another, the NNMi discovery process sends a notification to the `NmsApa` service. The `NmsApa` service uses this notification to recalculate the status of the connection.

## Device Generated Traps

**ColdStart and WarmStart traps** — The `NmsApa` service subscribes to notifications from the Events system for `ColdStart` and `WarmStart` traps. These notifications trigger the `NmsApa` service to initiate a rediscovery of device information from the node that generated the trap.

**LinkUp and LinkDown traps** — The `NmsApa` service subscribes to notifications from the Events system for `LinkUp` and `LinkDown` traps, as well as from some vendor-specific link traps it receives. These notifications trigger the `NmsApa` service to initiate a rediscovery of device information from the node that generated the trap.

For a complete list of the trap incident configurations that NNMi provides, see the NNMi help or select the **SNMP Trap Configuration** tab from the Incident Configuration view.

# NNMi Management Configuration Changes

During the span of a day, an NNMi tools administrator may complete several NNMi configuration changes. The following scenarios illustrate some common NNMi management configuration changes and how NNMi responds to these changes.

- **NNMi Administrator unmanages an IPv4 address or puts it out-of-service**

  The `NmsApa` service receives a notification from StatePoller after the `pingState` is set to `Not Polled`. The `NmsApa` service reacts to this notification and sets the status of the IPv4 address to `No Status`.

- **NNMi Administrator manages an IPv4 address or puts it back in service**

  The `NmsApa` service receives a notification from StatePoller after the `pingState` is set to the measured value. The `NmsApa` service reacts to this notification and calculates the status of the IPv4 address based upon the measured value.

- **NNMi Administrator unmanages an interface or puts it out-of-service**

  The `NmsApa` service receives a notification from StatePoller after the `operState` is set to `Not Polled`. The `NmsApa` service reacts to this notification and sets the status of the interface to `No Status`.

- **NNMi Administrator manages an interface or puts it back in service**

  The `NmsApa` service receives a notification from StatePoller after the `operState` is set to the measured value. The `NmsApa` service reacts to this notification and calculates the status of the interface based upon the measured value.

- **NNMi Administrator unmanages a node or puts it out-of-service**

  The `NmsApa` Service receives a notification from StatePoller after the `agentState` is set to `Not Polled`. `operState` is set to `Not Polled` for all interfaces and `pingState` is set to `Not Polled` for all IPv4 addresses. The `NmsApa` service reacts to this notification and sets the status of the node to `No Status`.

- **NNMi Administrator manages a node or puts it back in service**

  The `NmsApa` Service receives a notification from StatePoller after the `agentState` is set to the measured value. `operState` is set to the measured value for all interfaces and `pingState` is set to the measured value for all IPv4 addresses. The `NmsApa` service reacts to this notification and calculates the status of the node.

# Glossary

## A

### account

See user account.

### active cluster node

See active server.

### active server

The server currently running the NNMi processes in a high availability configuration.

### address hint

See discovery hint.

### ARP cache

The ARP (Address Resolution Protocol) cache is an operating system table that maps Data Link Layer (OSI Layer 2) addresses to Network Layer (OSI Layer 3) addresses. Data Link Layer addresses are typically MAC addresses, while Network Layer addresses are typically IP addresses. In rule-based discovery, NNMi uses ARP cache entries on discovered nodes (as well as other techniques) to find additional nodes that can be checked against the current discovery rules.

### auto-discovery

See rule-based discovery.

## C

### Causal Engine

NNMi technology that applies root cause analysis (RCA) to network symptoms, using a causality-based approach. Causal Engine RCA is triggered by certain occurrences, including changes detected as a result of state polling, SNMP traps, and specific incidents. The Causal Engine uses RCA to determine the status of managed objects, to formulate conclusions about them, and to generate root cause incidents.

### causality

Denotes the relationship between one event (the cause) and another event (the effect) which is the direct consequence (result) of the first. NNMi uses causality analysis algorithms to analyze event cycles and identify solutions for resolving network issues.

### cluster

In an NNMi context, a grouping of hardware and software, linked via high availability technology or via jboss clustering capabilities, that works together to ensure functional and data continuity in the event of component overload or failure. The computers in a cluster are commonly connected to each other through high speed LANs. Clusters are usually deployed to improve availability and/or performance.

### cluster member or node

In an NNMi context, a system within a high availability cluster that has been or will be configured to support NNMi high availability.

### community string

A password-like mechanism used in SNMPv1 and SNMPv2C implementations to authenticate SNMP queries to SNMP agents. The community string is passed in cleartext in SNMP packets, making it vulnerable to packet sniffing. SNMPv3 provides stronger security mechanisms for authentication.

### conclusion

In NNMi, supporting detail generated and used by the Causal Engine that sheds further light on how the Causal Engine determined status and root cause incidents for a managed object.

### console

See NNMi console.

## D

### discovery hint

An IP address found by NNMi via an SNMP ARP cache query; a CDP, EDP, or other discovery protocol

query; or a ping sweep. NNMi further queries IP addresses found as discovery hints, then checks the results against the current discovery rules in rule-based discovery.

### discovery process

The process by which NNMi gathers information about network nodes so that they can be placed under management. Initial discovery runs as a two-phase process, returning device inventory information and then network connectivity information.

After initial discovery, the discovery process is ongoing. In list-based discovery, this means devices in the list of seeds will be updated if their configuration changes. In rule-based discovery, new devices will also be added if they match current discovery rules. Discovery can also be initiated on demand for a device or set of devices from the NNMi console or from the command line.

See also spiral discovery, rule-based discovery, and list-based discovery.

### discovery rule

A range of user-defined IP addresses and/or system object IDs (OIDs) used to limit the rule-based discovery process. Configure discovery rules in the **Discovery Configuration** portion of the NNMi console under **Auto-Discovery Rules**. See also rule-based discovery.

### discovery seed

See seed.

## E

### embedded database

The database included with NNMi. NNMi can also be configured to use an external Oracle database instead of the embedded database for most of its tables. See also PostgreSQL.

### episode

A term used in NNMi root cause analysis to refer to a specific duration, triggered by a primary failure, during which secondary failures are suppressed or are correlated under the primary failure.

## F

### fault polling

A key NNMi monitoring activity, in which NNMi issues ICMP pings and/or SNMP read-only queries of status MIBs for its managed interfaces, IP addresses, and SNMP agents in order to determine the state of each managed object. Users can customize the types of fault polling performed for different interface groups, node groups, and nodes under **Monitoring Configuration** in the **Configuration** workspace of the NNMi console. Fault polling is a subset of state polling.

## H

### HA

See high availability.

### HA resource group

In modern high availability environments such as HP ServiceGuard, Veritas Cluster Server, or Microsoft Cluster Services, applications are represented as compounds of resources, such as the application itself, its shared file systems and a virtual IP address. The resources comprise an *HA resource group*, which represents an application running in a cluster environment.

### high availability

Used in this guide to refer to a hardware and software configuration that provides for uninterrupted service if part of the configuration fails. High availability (HA) means that the configuration has redundant components to keep applications running at all times even if a component fails. NNMi can be configured to support one of several commercially available HA solutions.

### HP Network Node Manager i Software

An HP software product (abbreviated NNMi) designed to aid network administration and to consolidate network management activities, including the ongoing discovery of network nodes, monitoring events, and network fault management. Primarily accessed via the NNMi console.

### HP Network Node Manager i-series Software

Also called HP NNM i-series Software, a family of HP software products built around the HP Network Node Manager i Software product (NNMi). HP NNM i-series Software includes NNMi, NNMi Advanced, and associated iSPIs such as the NNM iSPI for Performance and the NNM iSPI for MPLS.

## I

### ICMP

See Internet Control Message Protocol.

### incident

In NNMi, a notification of an occurrence related to your network, displayed in NNMi console incident views and forms. NNMi includes a number of **Incident Management** and **Incident Browsing** views that allow users to filter incidents based on incident attributes. Most incident views display incidents generated directly by NNMi (sometimes called *management events*). NNMi also includes views for browsing incidents generated from SNMP traps and from NNM 6.x/7.x events.

### interface

A physical port used to connect a node to the network.

### interface group

One of NNMi's primary filtering techniques, where interfaces are grouped together in order to apply settings to a group or filter visualizations by group. Interface groups can be used for any or all of the following: configuring monitoring, filtering table views, and customizing map views. See also node group.

### Internet Control Message Protocol

One of the core protocols of the Internet protocol suite (TCP/IP). ICMP ping is used by NNMi along with SNMP queries for state polling.

### iSPI

See NNM iSPI.

## J

### jboss application server

NNMi processes utilize a jboss application server called `ovjboss`. This application server is the main NNMi process, hosting the Java 2 Platform, Enterprise Edition (J2EE), and Enterprise Java Beans (EJB) business logic and database access capabilities. NNM iSPIs have their own jboss application servers.

### JGroups

An open source technology for multicast communication and clustering; used internally by jboss for its clustering capabilities.

## L

### L2

See Layer 2.

### L3

See Layer 3.

### Layer 2

Refers to the Data Link Layer of the multi-layered communication model, Open Systems Interconnection (OSI). The data link layer moves data across the physical links in the network. NNMi Layer 2 views provide information about the physical connectivity of devices.

### Layer 3

Refers to the Network Layer of the multi-layered communication model, Open Systems Interconnection (OSI). The network layer is concerned with knowing the address of the neighboring nodes in the network, selecting routes, and quality of service. NNMi Layer 3 views provide information about connectivity from a routing perspective.

### list-based discovery

A process, based on a list of seeds, that discovers and returns detailed network information *only about the nodes that you specify as seeds*. List-based discovery maintains a limited network inventory for specific

queries and tasks. Contrast with rule-based discovery. See also discovery process and spiral discovery.

### logical volume

A computer storage virtualization term referring to an arbitrarily sized space in a volume group that can be used as a separate file system or as a device swap space. Several of the high availability products supported by NNMi use logical volumes in their shared file systems.

## M

### management server

The NNMi management server is the computer system on which the NNMi software is installed. The NNMi processes and services run on the NNMi management server. (Prior NNM revisions used the term "NNM management station" for this system.)

### MIB

See Management Information Base.

### Management Information Base

In SNMP, the collection of data about the managed network, organized hierarchically. The data objects within the management information base refer to characteristics of managed devices. NNMi collects network management information by making SNMP queries to and receiving SNMP traps from managed nodes using MIB data objects (sometimes called "MIB objects," "objects," or "MIBs").

## N

### NNM 6.x/7.x events

An NNMi term for events forwarded from older NNM management stations to NNMi. NNMi provides incident views for browsing the incidents that NNMi generates from these forwarded events.

### NNM iSPI

A Smart Plug-in within the HP Network Node Manager i-series Software family. An NNM iSPI adds functionality to NNMi for a specific technology such as MPLS or for a specific domain such as network engineering.

### NNMi

See HP Network Node Manager i Software.

### NNMi console

The NNMi user interface. Operators and administrators use the NNMi console for network management tasks in NNMi.

### node

In the network context, a computer system or device (for example, printer, router, or bridge) in a network. While nodes that are able to respond to SNMP queries provide NNMi with the most comprehensive management information, NNMi can also perform restricted management of non-SNMP nodes.

### node group

One of NNMi's primary filtering techniques, where nodes are grouped together in order to apply settings to a group or filter visualizations by group. Node groups can be used for any or all of the following: configuring monitoring, filtering table views, and customizing map views. See also interface group.

## O

### OID

See Object Identifier.

### Object Identifier

In SNMP, a numerical sequence that identifies a MIB data object. An OID consists of numbers separated by dots in which each number represents a particular data object at that level of the MIB hierarchy. The OID is the numerical equivalent of the MIB object name, for example, the MIB object name `iso.org.dod.internet.mgmt.mib-2.bgp.bgpTraps.bgpEstablished` is equivalent to its OID `1.3.6.1.2.1.15.0.1`.

### ovstatus command

A command that reports the current status of the NNMi managed processes. Can be invoked from the NNMi console (**Tools** → **NNM Status**) or at a command prompt. Refer to the *ovstatus* reference page, or the UNIX manpage.

### ovstart command

A command that starts the NNMi managed processes. Invoked at a command prompt. Refer to the *ovstart* reference page, or the UNIX manpage.

### ovstop command

A command that stops the NNMi managed processes. Invoked at a command prompt. Refer to the *ovstop* reference page, or the UNIX manpage.

## P

### ping sweep

A network probe technique that sends ICMP ECHO requests to multiple IP addresses in order to determine which addresses are assigned to responsive nodes. When enabled in rule-based discovery, NNMi can use ping sweep on configured IP address ranges to find additional nodes. Some network administrators block ICMP ECHO requests because ping sweeps can be used in denial-of-service attacks.

### port

In a network hardware context, a connector for passing information into and out of a network device.

### PostgreSQL

An open source relational database which NNMi uses by default to store information such as topology, incidents, and configuration information. NNMi can also be configured to use Oracle instead of PostgreSQL for most of its tables.

### public key certificate

Used in network security and encryption, a file that incorporates a digital signature to bind together a public key with identity information. A certificate is used to verify that a public key belongs to an individual or organization. NNMi uses SSL certificates, which contain a public key and a private key, for authentication and encryption of client/server communication.

## R

### RCA

See root cause analysis.

### region

In NNMi, a grouping of devices for the purpose of configuring communication settings such as timeout values and access credentials.

### role

See user role.

### rule

See discovery rule.

### rule-based discovery

Often called *auto-discovery*, NNMi can use rule-based discovery to seek out nodes that NNMi should add to its database, following user-specified discovery rules. NNMi looks for discovery hints in data from discovered nodes, then checks these candidates against the specified discovery rules. Configure discovery rules in the **Discovery Configuration** portion of the NNMi console under **Auto-Discovery Rules**. Contrast with list-based discovery.

### root cause analysis

In NNMi, root cause analysis (RCA) refers to a class of problem solving methods used by NNMi to determine root causes for network issues. In NNMi, the root cause is the actionable issue that will resolve associated problem symptoms if it is addressed. NNMi uses the identification of the root cause in two key ways: to notify the user of the actionable problem and to suppress reporting of secondary problem symptoms until the root cause issue has been resolved. Determination of root cause may result in status changes for managed objects, generation of root cause incidents, or both.

A example of how NNMi uses RCA is the scenario in which a managed router fails, and managed nodes on the other side of the router from the NNMi management server can no longer respond to state polling queries. NNMi uses RCA to determine that the state polling failures are secondary problem symptoms. It reports the router failure as the root cause incident and refrains from reporting the problem symptoms for the downstream nodes until the root cause router failure is resolved.

### root cause incident

An NNMi incident in which the *Correlation Nature* attribute is set to *Root Cause*. NNMi uses root cause analysis (RCA) to establish the root cause incident as the actionable issue that will resolve associated problem symptoms if it is addressed. See root cause analysis.

## S

### seed

A network node that helps NNMi discover your network by acting as a starting point for the network discovery process. For example, a seed might be a core router in your management environment. Each seed is identified by an IP address or host name. Unless rule-based discovery has been configured,

NNMi's discovery process is limited to list-based discovery of specified seeds.

### seeded discovery

See list-based discovery.

### Simple Network Management Protocol

A simple protocol operating at the application layer (Layer 7) of the OSI model, by which management information for a network element can be inspected or altered by remote users. SNMP is the predominant protocol used by NNMi to exchange network management information with agent processes on managed nodes. NNMi supports the three most common versions of SNMP: SNMPv1, SNMPv2C, and SNMPv3.

### SNMP

See Simple Network Management Protocol.

### SNMP trap

Network management via polling (solicited responses from SNMP agents) is an SNMP design principle that promotes simplicity. However, the protocol does provide for communication of unsolicited messages from SNMP agents to the SNMP manager process (in this case, NNMi). Unsolicited agent messages are known as "traps" and are generated by SNMP agents in response to internal state changes or fault conditions. NNMi generates incidents from received SNMP traps, displayed in the **SNMP Traps** incident browsing view.

### SNMP trap storm

A high number of unsolicited SNMP agent messages that can overwhelm an SNMP manager process (in this case, NNMi). You can configure SNMP trap storm thresholds in NNMi, using the `nnmtrapconfig.ovpl` script. NNMi blocks traps when incoming trap rates exceed the specified threshold rate, until the trap rates fall below the re-arm rate.

### spiral discovery

NNMi's ongoing refinement of network topology information, which includes information about inventory, containment, relationships, and connectivity in networks managed by NNMi. See also discovery process, rule-based discovery, and list-based discovery.

### state

NNMi generally uses the term **state** for self-reported managed object responses related to MIB II `ifAdminStatus`, MIB II `ifOperStatus`, performance, or availability. Contrast with status.

### state polling

The directed monitoring performed by NNMi's State Poller, which uses ICMP ping and SNMP queries to retrieve fault, performance, component health, and availability data from managed objects. See also fault polling.

### status

In NNMi, an attribute of a managed object that indicates its overall health. The status is calculated by the Causal Engine from the managed object's outstanding conclusions. Contrast with state.

### sysObjectID

See system object ID.

### system object ID

In NNMi, a specialized term for an SNMP Object Identifier that identifies a model or type of network element. The system object ID is part of a network element's MIB object, which is queried by NNMi from individual nodes during discovery. Examples of network element types that can be classified by their system object IDs include any member of the HP ProCurve switch family, an HP J8715A ProCurve Switch, and an HP SNMP agent for HP IPF systems. Other vendors' network elements can be likewise classified according to their system object IDs. A key use for the system object ID is in defining NNMi Device Profiles, which specify characteristics of network elements that can be deduced once a network element's type is known.

### system account

In NNMi, a special account provided for use during NNMi installation. After installation, the NNMi system account should only be used for command-line security and for recovery purposes. Contrast with user account.

## T

### topology (network)

In communication networks, a schematic description of the arrangement of a network, including its nodes and connections.

**trap**

See SNMP trap.

## U

**unconnected interface**

From NNMi's perspective, an unconnected interface is an interface that is not connected to another device discovered by NNMi. By default, the only unconnected interfaces that NNMi monitors are those that have IP addresses *and* are contained in nodes from the **Routers** node group.

**user account**

In NNMi, a way to provide access to NNMi for users or groups of users. NNMi user accounts are set up in the NNMi console and implement predetermined user roles. See system account and user role.

**user role**

As part of setting up user access, the NNMi administrator assigns a pre-configured user role to each NNMi user account. User roles determine which user accounts have access to the NNMi console, as well as which workspaces and actions are available to each user account. NNMi provides the following hierarchical user roles, which are predefined by the program and cannot be modified: *Administrator*, *Web Service Client*, *Operator Level 2*, *Operator Level 1*, *Guest*. See also user account.

## V

**virtual host name**

The host name associated with a virtual IP address.

**virtual IP address**

An IP address that is not tied to any particular network hardware, used in high availability configurations to send uninterrupted network traffic to the most appropriate server based on current failover or load-balancing needs.

**volume group**

A computer storage virtualization term referring to one or more disk drives that are configured to form a single large storage area. Several of the high availability products supported by NNMi use volume groups in their shared file systems.

# Index


## A

About HP Network Node Manager i-series window, 68

access credentials
- community strings, 36
- configuring, 37
- multiple, 41

Actions menu, 183

active
- protocols, 40

adding NNM 6.x/7.x
- management stations, 176
- views, 179

address, management server
- preferred, 38
- verifying, 43

address, NNMi management server
- changing, 132

AddressNotResponding incident, 246

advantages
- list-based discovery, 47
- rule-based discovery, 47

agent, SNMP
- responsive, 248
- status, 244
- unresponsive, 247

AlarmPoint
- agent, 223
- client, 223
- documentation, 223
- integration
  - disabling, 224
  - enabling, 223
  - overview, 221 to 223
  - troubleshooting, 224
  - usage, 223
- Java Client, 223
- servers, 223
- supported versions, 222

alarms, NNM 6.x/7.x
- categories, 177
- launching dynamic views, 183
- mapping, 177 to 178

alerting application, 221

analyzing managed nodes, 243

APAgent (AlarmPoint), 223

APClient (AlarmPoint), 223

application
- alerting, 221
- server, AlarmPoint, 223

Application_A.log file, 121

Application Alert Alarms category, NNM 6.x/7.x, 178

Application Status incident category, 178

architecture, HA cluster, 79

archive files, 126

ARP cache, 51

attribute
- Author, 28
- Ordering, 28

authentication failures, reducing, 43

Author attribute, 28

## B

backbone, network, 45

backing up
- data
  - complete backup, 127
  - embedded database only, 128
  - script, 125 to 127
  - strategy, 123 to 124

backup
- file, 129
- script
  - backing up data, 125
  - restoring data, 127

Basic shared disk format, 85, 93

batching monitoring configuration, 64

behavior, out-of-box polling, 59

benefits
list-based discovery, 47
rule-based discovery, 47

best practices
Author attribute, 28
batching monitoring configuration, 64
creating
object group definitions, 62
reusable node groups, 63
double-checking order numbers, 41
No Device Profile, 152
Ordering attribute, 28
overview, 15
preferred management address, 38
preparing NNMi configuration move, 131
saving existing configuration, 28
short polling intervals, 63
verifying order numbers, 65

BIND, 100

browsers, incident, 173

## C

cache, ARP, 51

categories
mapping, 177
status, 244

Causal Engine
concepts, 242
features, 241
network failure scenarios, 246

cause, underlying, 242

certificate, public key, 73 to 74

changes, configuration
management, 268
network, 267

changing NNMi management server
IP address, 132
overview, 131

checklist, polling, 58

CIAs, 179

Cisco
routers
defining node groups, 30
hierarchies, 31
switches
defining node groups, 30
hierarchies, 31

clearing shadow, 262

Client, AlarmPoint Java, 223

cluster.exe command, 97

cluster.log file, 120

cluster architecture for HA, 79

cluster nodes, configuring HA
NNMi
primary, 86 to 88
secondary, 88
NNM iSPI for Performance
primary, 93 to 94
secondary, 94

cmcluster file, 120

coexistence, HP Performance Insight, 23

collecting data
status polling
choosing data, 64
verifying, 67

command-line options
nnmbackup.ovpl, 125
nnmbackupembdb.ovpl, 129
nnmrestore.ovpl, 128
nnmrestoreembdb.ovpl, 129

commands
cluster.exe, 97
hostname, 100
keytool, 74
netstat, 74
nnm.envvars.bat, 236 to 238
nnm.envvars.sh, 236
nnmcommconf.ovpl, 42
nnmconfigexport.ovpl
outputting configuration to XML, 132
saving management station configuration, 176
nnmconfigimport.ovpl, 132
nnmdatareplicator.ovpl, 97
nnmdumpevents, 183
nnmhaconfigure.ovpl, 116
nnmhadisk.ovpl
configuration file replication, 97
troubleshooting nmsdbmgr, 116
nnmhargconfigure.ovpl, 114
nnmlicense.ovpl, 98 to 100
changing management server, 132
nnmloadseeds.ovpl, 53
nslookup, 84, 92
ovaddr, 179
ovspmd
troubleshooting, 115
ovstart
troubleshooting, 115
ovstop
troubleshooting, 115
without causing failover, 102, 111
ovtopodump, 181
ovtopofix, 181
ping, 247 to 254
remsh, 78
sendMsg.ovpl, 181 to 182
snmptrap, 182
ssh, 78
traceroute, 40
xnmevents, 175

communicating https connection information to NNMi users, 76

communication
concepts, 35
configuration regions, 39
configuring, 41
evaluating configuration, 41
planning, 39
settings, 42
tuning, 43
verifying settings, 43

community strings
access credentials, 36
multiple, 41

Component Health monitoring, 64 to 69

component registration files, 126

concepts
Causal Engine, 242
communication, 35
configuration, 27
discovery, 138
event monitoring, 144
HA, 79
state polling, 57 to 58
status monitoring, 141

Configuration
incident category, 178
workspace, 179

configuration
batching monitoring, 64
changes
management, 268
network, 267
comparisons
customizing event monitoring, 142
network discovery, 136
status monitoring, 139
concepts, 27
files
HA clusters, 118
HA configuration, 118
replication for HA, 97
HA information
NNMi, 84
NNM iSPI for Performance, 91
HPOM integration, 198 to 203
incident category, 178
levels, 37
moving NNMi, 132
nodes, 40
polling example, 59
preparing NNMi move, 131
redoing, 33
regions, 39
saving
existing, 28
management station, 176
scope data, 126
scripts, HA clusters, 118
settings, 33
shared data, 126
Tomcat, 73
transaction-based updates, 28
troubleshooting HA, 113 to 117
verifying network monitoring, 66 to 67
workspace, 179

Configuration Alarms category, NNM 6.x/7.x, 178

Configuration workspace
configuring state polling, 64
evaluating state polling, 66

configuring
access credentials, 37
communication, 41
HA cluster nodes, primary
NNMi, 86 to 88
NNM iSPI for Performance, 93 to 94
HA cluster nodes, secondary
NNMi, 88
NNM iSPI for Performance, 94
HA clusters
NNMi, 84 to 88
NNM iSPI for Performance, 91 to 95
interface
groups, 64
monitoring, 65
list-based discovery, 53
NNM 6.x/7.x
adding views, 179
forwarding events, 174
node
groups, 64
monitoring, 65
retries, 37
rule-based discovery, 53
shared disk, 97
SNMP access, 148 to 152
state polling, 64 to 66
timeouts, 37

connection
communicating https information, 76
down
operationally, 255
router, 264
up
operationally, 256
router, 264

ConnectionDown incident, 246

ConnectionPartiallyUnresponsive incident, 246

connectivity, verifying network, 54

connector blocks, https configuration, 76

console. *See* NNMi console

containment, node group, 31

controlling traffic, 40

CPU resources, 69

creating
NNM 6.x/7.x management station entity, 178
object group definitions, 62
reusable node groups, 63
shadow, 261

credentials, access
community strings, 36
configuring, 37
multiple, 41

custom incident attributes, 179

customizing
event monitoring, 142

## D

data
backing up
complete backup, 127
embedded database only, 128
script, 125 to 127
strategy, 123 to 124
collection, State Poller, 64
configuration scope, 126
event scope, 127
restoring
embedded database only, 128
script, 127
shared, configuration, 126
shared disk, 96
topology scope, 127
verifying collection, 67

database
backing up embedded, 128
resetting, 33
restoring embedded, 128
topology, 57

defaults
community strings, 43
configuration settings, 33
discovery, 45
https ports, 76
meaningful, communication, 39
routers, 56
rule-based discovery, 53
switches, 56
verifying settings, 66

defining node groups, 30

deleting
discovered nodes, 54
license key, 132

destination list file, 175

details, node, 180

devices
filters, 32
generated traps, 267
profiles
concepts, 32
State Polling, 63
SNMP access, 42

directories. *See* locations

disabling
AlarmPoint integration, 224
network access, 38
nGenius Performance Manager integration, 228
SNMP, 40
traffic, 38

disabling HPOM integration, 194

disadvantages, discovery, 47

discovery
approaches
list-based discovery, 47
rule-based discovery, 47 to 48
deleting nodes, 54
disadvantages, 48
evaluating
list-based discovery, 54
rule-based discovery, 55 to 56
key concepts, 138
NNM 6.x/7.x
comparison, 136
migration, 152 to 161
performance, 43
restarting, 33
routers, 55
spiral, 45
switches, 55
transaction-based updates, exception, 28

disk
failover, 116
group, HA configuration, 85, 93
shared
copying data files, 84
directories, 96

DiskGroup_A.log file, 121

DNS, 100

documentation
AlarmPoint, 223
hardware and software requirements, 21
locations, 145
nGenius Performance Manager, 227
related, 18

Domain Name Service, 100

down
connection, 255
interface
administratively, 253
operationally, 251
node
creating shadow, 261
distribution router, 257
router, 264

dynamic disks, 98

dynamic views
launching from NNMi, 183
NNM 6.x/7.x, 178

## E

editing trapd.conf, 176

embedded database
backing up and restoring, 128
relational, 123

enabling
AlarmPoint integration, 223
HPOM integration
UNIX, 191
Windows, 190
https protocol, 73
nGenius Performance Manager integration, 227
rule-based discovery, 55

end nodes, avoiding managing, 45

end-to-end diagnosis, 241

environment, test, 15

environment variables
overview, 235
UNIX
administration, 237
installation, 16
Windows
administration, 236 to 238
installation, 16

episodes, 243

Error Alarms category, NNM 6.x/7.x, 178

/etc/hosts, 100

evaluating
communication configuration, 41
discovery configuration, 54
state polling configuration, 66 to 68

evaluation order, 58

Event Configuration window, NNM 6.x/7.x, 174

Event Configurator window, NNM 6.x/7.x, 175, 183

events
forwarding
management server, 174
verifying, 181
generating node/interface down/up, 181
migrating from NNM 6.x/7.x, 167 to 172
monitoring
concepts, 144
customizing, 142
OV_Message, NNM 6.x/7.x, 182
reducing, 176
scope data, 127
systems as traps, 242

examples
configuring node groups, 65
HPOM integration incident filters, 202
polling configuration, 59
SNMP information, 148

excluding devices from discovery, 48

extended monitoring, 59

external relational database, 123

## F

failover, disk, 116

failure
network scenarios, 246 to 264
reducing authentication, 43

files
archive, 126
backing up, 123
backup, 129
component registration, 126
destination list, 175
HA configuration
files, 118
log files, 120
scripts, 118
ipNoLookup.conf, 151
jbossServer.log, 116
license.txt, 100
management server, 132
LicFile.txt, 86, 98
backup, 127
netmon.cmstr, 148
nGeniusNNM8.zip, 227
nmsdbmgr.log, 120
nnmdatareplicator.conf, 118
not updating on cluster node, 113
oid_to_sym, 152
OLDsyslog.log, 120
ov.conf, 116
HA configuration, 118
ovspmd.log, 120
postgres.log, 120
replication, 97
<resource_group>.cntl.log, 120
server.xml, 74 to 76
shared disk, 117
snmpout.txt, 149
syslog.log, 120
system type, 85, 92
tar, 128
trapd.conf, 176
version information, 126
XML, 132

file system type, HA configuration, 85, 93

filtering
interface groups, 33
node groups, 29

filters
configuring node level, 176
device, 32
incident, 200 to 202

firewall, troubleshooting HPOM integration, 197

firewalls
controlling traffic, 40
disabling network access, 38

flow model, task, 27

forms
Incident Configuration, 177
Interface Group, 61
Interface Settings, 69
Management Stations, 179
Monitoring Configuration, 57, 69
Node Group, 61
Node Settings, 69

forwarded incidents, troubleshooting HPOM, 195 to 197

FORWARD field, NNM 6.x/7.x event configuration, 176

forwarding, event
NNM 6.x/7.x to NNMi management server, 174
verifying, 181

forwarding incidents to HPOM, 188

## G

generating
node/interface down/up events, 181
traps, 267

getting public key certificate, 73 to 74

group
disk, 85, 93
volume, 85, 93

groups
interface
filtering, 33
purpose, 29
nodes, 29

GUI. *See* NNMi console

## H

HA_nnmhaserver.log file
UNIX, 120
Windows, 120

HA clusters
architecture, 79
changing hostnames
NNMi, 99 to 102
NNM iSPI for Performance, 103
changing IP addresses
NNMi, 99 to 102
NNM iSPI for Performance, 103
concepts, 79
configuration
files, 118
information, NNMi, 84
information, NNM iSPI for Performance, 91
log files, 120
manpages, 83
reference pages, 83
scripts, 118
configuring
add-on iSPIs, 89
NNMi, 84 to 88
NNM iSPI for Performance, 91 to 95
licenses, 98
maintaining
add-on iSPIs, 102
NNMi, 99 to 102
NNM iSPI for Performance, 102 to 103
maintenance mode, 99
prerequisites, 78
scenario, 80
shared data, 96 to 97
startup problems
nmsdbmgr, 116
NNMi, 115
pmd, 116
supported products, 78
terms, 80
troubleshooting configuration, 113 to 117
unconfiguring
NNMi, 104 to 107
NNM iSPI for Performance, 107 to 109

HA configuration, 118

haconfigure.log file
UNIX, 120
Windows, 120

hardware
required, 19
supported, 21

HA resource group
cannot start, 114
configuration
NNMi, 84
NNM iSPI for Performance, 92
description, 80
stopping
NNMi, 105
NNM iSPI for Performance, 108

hierarchies, node group, 31

high availability. *See* HA clusters; HA resource group

hints
IP address ranges, 55
system ID ranges, 56

Home Base, NNM 6.x/7.x, 183

host, virtual for HA configuration
NNMi, 84
NNM iSPI for Performance, 92

hostname command, 100

hostnames, changing for HA
NNMi, 99 to 102
NNM iSPI for performance, 103

HPOM integration
disabling, 194
documentation, 189
enabling
UNIX, 191
Windows, 190
overview, 187
supported versions, 189
troubleshooting, 195 to 197
usage, 193
value, 188

HP Operations Manger. *See* HPOM integration

HP Performance Insight, 23

HP ServiceGuard
HA resource group, 80
HP-UX, 78
Linux, 78

HP-UX
commands, 114
configuring shared disks, 97
HP ServiceGuard, 78
log files, 120
NIS, 100
nnmharg.ovpl script, 119
shared disk format, 85, 93
virtual hosts, 85, 92

http protocol, 73

https
default port, 76
protocol, 74

## I

ICMP
address monitoring, 63
disabling traffic, 38
IPv4Address
responsive, 250
unresponsive, 249
protocol, 35
requests, 43
State Poller, 38

Incident Configuration form, verifying configurations, 177

incidents
browsers, 173
categories, 177 to 178
definitions, 246
filters, 200 to 202
forwarding, 188
verifying NNM 6.x/7.x in NNMi console, 177

Incidents workspace, 181

Incident Web Service, HPOM, 188

initiating status poll, 67

installing
AlarmPoint Java Client, 223
HP Performance Insight, 23
Java Runtime Environment, 178
license for HA cluster, 98
public key certificates, 73 to 74

integrating AlarmPoint with NNMi
disabling, 224
enabling, 223
overview, 221 to 223
troubleshooting, 224
usage, 223

integrating nGenius Performance Manager with NNMi
disabling, 228
enabling, 227
troubleshooting, 226 to 228
usage, 227

integrating NNM 6.x/7.x with NNMi
event forwarding, 174 to 178
remote view launching, 178 to 180
testing, 180 to 183
troubleshooting, 183

integrating NNMi with HPOM
disabling, 194
documentation, HPOM, 189
enabling
UNIX, 191
Windows, 190
overview, 187
supported versions, 189
troubleshooting, 195 to 197
usage, 193
value, 188

integration module, HPOM, 188

interface
configuring monitoring, 65
down
administratively, 253
operationally, 251
groups
configuring, 64
filtering, 33
preconfigured, 62
purpose, 29
verifying, 66
model, 28
monitoring, 59
moving, 267
settings, 33
status, 244
up
administratively, 254
operationally, 252
virtual host network for HA configuration, 85, 92

InterfaceDown incident, 246

Interface Group form, 61

Interface Groups workspace, 61

Interface Settings form, 69

IP address
changing for HA
NNMi, 99 to 102
NNM iSPI for Performance, 103
changing NNMi management server, 132
ranges, 53 to 55

ipNoLookup.conf file, 151

IPv4 address, 244

IPv4Address to ICMP
responsive, 250
unresponsive, 249

iSPIs
NNM iSPI for Performance
Component Health, 64

IWS, HPOM, 188

## J

Java Client, AlarmPoint, 223

Java Runtime Environment, installing, 178

jboss
directories, 126
versions, 74

jbossServer.log file
HP-UX, 120
Linux, 121
Solaris, 121
UNIX, 116
Windows, 116, 120

## K

key, public, 73 to 74

keytool command, 74

## L

lag time, testing, 40

latency, network, 36

Launcher, NNM 6.x/7.x, 183

launching NNM 6.x/7.x dynamic views, 183

levels, configuration, 37

license.txt file, 100
management server, 132

licenses
capacity, 46
directory locations, 127
HA clusters, 98
limitations, 48
purchasing additional node pack, 54

LicFile.txt file
backing up, 86
NNMi data, 127
updating licenses.txt, 98

limited environment, 15

limiting name resolution, 151

Linux
commands, 114
configuring shared disks, 97
HPOM integration, 189, 214
HP ServiceGuard, 78
log files, 120
NIS, 100
nnmharg.ovpl, 119
shared disk format, 85, 93
virtual hosts, 85, 92

list-based discovery, 54
evaluating, 54
overview, 47

locations
documentation, 145
AlarmPoint, 223
nGenius Performance Manager, 227
environment variables
administration, 236 to 238
installation, 16
ipNoLookup.conf, 151
jboss deployment directory, 126
LicFile.txt file, 127
netmon.cmstr, 148
nGeniusNNM8.zip, 227
SNMP MIB, 127
trapd.conf, 176

log files, HA cluster
configuration, 120
not updating, 113

lvm2 shared disk format, 85, 93

## M

maintaining HA clusters, 99 to 103

maintenance mode for HA, 99

managed nodes, analyzing, 243

management
address
preferred, 38
verifying, 43
configuration changes, 268
server
forwarding NNM 6.x/7.x events, 174
installing AlarmPoint Java Client, 223
moving NNMi, 131
server, disabling HPOM integration, 194
station
adding NNM 6.x/7.x to topology, 176
creating NNM 6.x/7.x entities, 178
saving configuration, 176

Management Stations form, 179

manpages, HA, 83

manually editing trapd.conf, 176

“many-to-many” arrangement, 187

mapping categories, 177

meaningful defaults for communication configuration, 39

membership, node group, 31

memory resources, 69

messages* log files
Linux, 120
Solaris, 121

MIB browser
example URL, 180
NNM 6.x/7.x, 183

MIB II variables, 64

Microsoft Cluster Services, 78
dynamic disks, 98
HA resource group, 80
nnmhamscs.vbs script, 119

migrating from NNM 6.x/7.x
controlling rate of migration, 173
discovery, 152 to 161
events, 167 to 172
options, 146 to 147
phases, 146
status monitoring, 161 to 166
suggested path, 145

migrating SNMP information, 148 to 152

MINCAUSE algorithm, 241

model
task flow, 27
user interface, 28

monitoring
configuration
batching, 64
settings, 33
verifying, 66 to 67
configuring node, 65
events
concepts, 144
customizing, 142
extended, 59
interface, 65
networks
nodes, 69
verifying settings, 66
settings, 67
status
concepts, 141
configuring, 139

Monitoring Configuration form
description, 57
setting polling types and intervals, 61
tuning status polling, 69

Mount_A.log file, 121

mount point, HA configuration, 85, 93

moving
interfaces, 267
NNMi
configuration, 132
management server, 131

MTTR, 226

## N

name resolution, limiting, 151

netmask, virtual host for HA configuration, 84, 92

netmon.cmstr file, 148

NetScout Systems nGenius Performance Manager. *See* nGenius Performance Manager

netstat command, 74

network configuration changes, 267

Network Information Service, 100

network interface, virtual host for HA configuration, 85, 92

network latency, 36

network operations center, 149

networks
backbones, 45
disabling access, 38
discovering
evaluating, 54 to 56
NNM 6.x/7.x, 136
failure scenarios, 246 to 264
loads, 69
monitoring
behavior, 59
nodes, 69
settings, 66
verifying
connectivity, 54
settings, 66

nGeniusNNM8.zip file, 227

nGenius Performance Manager
documentation, 227
integration
disabling, 228
enabling, 227
troubleshooting, 226 to 228
usage, 227
supported versions, 226

NIS, 100

NmsApa service
Causal Engine, 241
configuration changes, 268
device-generated traps, 267
network connections, 267
node
unreachable, 263
updated, 267
setting object status, 242

nmsdbmgr.log file
HP-UX, 120
Linux, 121
Solaris, 121
Windows, 120

nmsdbmgr service, 129
disk failover, 116
startup problems, 116

NNM 6.x/7.x
configuring event node list, 176
customizing event monitoring, 142
dynamic views, 178
integrating with NNMi
event forwarding, 174 to 178
remote view launching, 178 to 180
testing, 180 to 183
troubleshooting, 183
management station, 181 to 182
migrating to NNMi
discovery, 152 to 161
events, 167 to 172
options, 146 to 147
phases, 146
SNMP, 148 to 152
status monitoring, 161 to 166
suggested path, 145
network discovery, 136
status monitoring, 139
views, 178

NNM 6.x/7.x Events view, 183

nnm.envvars.bat command, 236 to 238

nnm.envvars.sh command, 236

nnmbackup.ovpl
command-line options, 125
configuring primary cluster node, 86
description, 124

nnmbackupembdb.ovpl
command-line options, 129
description, 124

nnmcommconf.ovpl command, 42

nnmconfigexport.ovpl
outputting configuration to XML, 132
saving management station configuration, 176

nnmconfigimport.ovpl command, 132

nnmdatareplicator.conf file, 118

nnmdatareplicator.ovpl
- command, 97
- script, 119

nnmdumpevents command, 183

nnmhaclusterinfo.ovpl script, 118, 119

nnmhaconfigure.ovpl
- command, 116
- script, 119

nnmhadisk.ovpl command, 117
- configuration file replication, 97
- troubleshooting nmsdbmgr, 116

nnmhadisk.ovpl script, 119

nnmhamonitor.ovpl script, 119

nnmhamscs.vbs script, 119

nnmharg.ovpl script, 119

nnmhargconfigure.ovpl
- command, 114
- script, 119

nnmhastart.ovpl script, 119

nnmhastartrg.ovpl script, 119

nnmhastop.ovpl script, 119

nnmhaunconfigure.ovpl script, 119

NNMi console
- accessing, 188
- HPOM, 188
- securing, 73 to 76
- transaction-based updates, 28
- verifying NNM 6.x/7.x incident configuration, 177

NNMi Quick Start Configuration wizard, 15

NNM iSPI for Performance
- Component Health, 64

nnmlicense.ovpl command, 98 to 100, 132
- changing management server, 132

nnmloadseeds.ovpl command, 53

nnmrestore.ovpl
- command-line options, 128
- description, 124

nnmrestoreembdb.ovpl
- command-line options, 129
- description, 124

NOC, 149

node/interface down/up events, 181

NodeDown incident, 246

Node Group form, 61

node groups
- configuring, 64
- defining, 30
- device filters, 32
- filtering, 29
- hierarchies, 31
- interface groups, 33
- membership, 31
- Non-SNMP Devices, 60
- preconfigured, 63
- status, 245
- verifying, 66

Node Groups workspace, 61

NodeOrConnectionDown incident, 246

nodes
- analyzing managed, 243
- configuration, 40
- configuring monitoring, 65
- configuring primary for HA clusters
  - NNMi, 86 to 88
  - NNM iSPI for Performance, 93 to 94
- configuring secondary for HA clusters
  - NNMi, 88
  - NNM iSPI for Performance, 94
- deleting discovered, 54
- details, 180
- down
  - connection, 264
  - creating shadow, 261
  - distribution router, 257
- filtering levels, 176
- monitoring, 69
- reachable, 263
- settings, 33
- SNMP configuration, 42
- status, 244
- unreachable, 263
- up
  - clearing shadow, 262
  - distribution router, 258
  - router, 264
- updating, 267

Node Settings form, 69

Nodes inventory view, 176

No Device Profile, 152

Non-SNMP Devices node group, 60

Nortel
- routers
  - defining node groups, 30
  - hierarchies, 31
- switches
  - defining node groups, 30
  - hierarchies, 31

nslookup command, 84, 92

## O

object groups, creating definitions, 62
objects, setting status, 242
offline backups, 125 to 126
oid_to_sym file, 152
OLDsyslog.log file, 120
OM. *See* HPOM integration
online backups, 125 to 126
OpenView Application Server, NNM 6.x/7.x, 179
Operations Manager. *See* HPOM integration
options, migrating from NNM 6.x/7.x, 146 to 147
Oracle Database and HA, 95
order, evaluation, 58
Ordering attribute
    auto-discovery rules, 48
    best practices, 28
ordering auto-discovery rules, 48
order numbers, verifying, 65
out-of-box polling behavior, 59
ov.conf file, 116, 118
OV_Message event, NNM 6.x/7.x, 182
ovaddr command, NNM 6.x/7.x, 179
ovalarm example URL, 180
OV Application Server, NNM 6.x/7.x, 179
ovas, 179
ovas, NNM 6.x/7.x, 179
ovdumpevents command, NNM 6.x/7.x, 183
ovet*.log file
    HP-UX, 120
    Linux, 121
    Solaris, 121
    Windows, 120
OV jovw example URL, 180
OV Launcher example URL, 180
ovspmd.log file
    HP-UX, 120
    Linux, 120
    Solaris, 121
    Windows, 120
ovspmd service
    troubleshooting, 115
ovstart command
    troubleshooting, 115
ovstop command
    troubleshooting, 115
    without causing failover, 102, 111
ovtopodump command, NNM 6.x/7.x, 181
ovtopofix command, NNM 6.x/7.x, 181
ovw, NNM 6.x/7.x, 183
ovwId, node details using, 180

## P

pages, HA man (UNIX), 83
pages, HA text files (Windows), 83
parameters
    HP UCMDB integration configuration, 217 to 219
    HPOM integration configuration, 198 to 202
partial backups, 125
password, permanent license, 100
paths, migrating from NNM 6.x/7.x, 145
PCs, avoiding managing, 45
performance, status polling, 67 to 68
Performance incident category, 178
Performance Insight, 23
permanent license password, 100
personal computers, avoiding managing, 45
phases, migrating from NNM 6.x/7.x, 146
ping, 49
    command, 247 to 254
    protocol, 35
    requests, 64
planning
    communication, 39
    polling intervals, 63
    state polling, 58 to 64
pmd startup problems, 116
point, mount, 85, 93
polling
    checklist, 58
    configuration example, 59
    out-of-box behavior, 59
    planning intervals, 63
    protocols, 38
    status
        evaluating performance, 67 to 68
        initiating, 67
        tuning, 68 to 69
port number, web server, 179

postgres.log file
HP-UX, 120
Linux, 120
Solaris, 121
Windows, 120

preconfigured
interface groups, 62
node groups, 63

preferences
management address, 38
SNMP version, 36

preparing NNMi configuration move, 131

prerequisites
hardware, 21
NNMi for HA cluster, 78
software, 21

primary HA cluster node
configuration information, 84, 91
configuring
NNMi, 86 to 88
NNM iSPI for Performance, 93 to 94

printers, avoiding managing, 45

problems, HA startup
nmsdbmgr, 116
NNMi, 115
pmd, 116

products, HA, 78

profiles, device
concepts, 32
State Polling, 63

protocol
active, 40
communication, 35
http, 73
polling, 38
SNMP, 243

public key certificate, installing, 73 to 74

purchasing additional node pack licenses, 54

## Q

Quick Start Configuration wizard, 15

## R

ranges, IP addresses, 55

reachable nodes, 263

redoing configuration, 33

reducing
authentication failures, 43
default community strings, 43
events, 176

re-enabling NNMi for HA clusters, 115

reference pages, HA, 83

regions, configuration, 39

related documentation, 18

relational database, backing up, 123

release notes, 21

removing license key, 132

remsh command, 78

replication, configuration file, 97

Report Presenter example URL, 180

requirements, hardware and software, 21

resetting
configuration, 33

resolution, limiting name, 151

<resource_group>.cntl.log file
HP-UX, 120
Linux, 120

resource group, HA, 84, 92

resources, system, 69

responsive
IPv4Address to ICMP, 250
SNMP agent to queries, 248

restarting
discovery, 33
NNMi after HA maintenance, 102
NNM iSPI for Performance after HA maintenance, 103

restore script, 127

restoring data
embedded database, 128
script, 127

retries
configuring, 37
tuning, 43
values, 40

root cause
generating conclusions, 242

routers
defaults, 56
defining node groups, 30
discovering, 55
hierarchies, 31
monitoring, 59

rule-based discovery
enabling, 55
evaluating, 55 to 56
overview, 47

rules, auto-discovery
evaluating, 55 to 56
ordering, 48

## S

saving management station configuration, 176

scenarios
HA cluster, 80
network failure, 246 to 264

scope
configuration, 126
event, 127
topology, 127

-scope all, 127

-scope config, 126

-scope event, 127

-scope topology, 127

scripts
backup, 125 to 127
HA configuration, 118
nnmhaclusterinfo.ovpl, 118
restoring data, 127

secondary HA cluster nodes
configuration information, 84
configuring
NNMi, 88
NNM iSPI for Performance, 94

secure socket layer technology, 73

securing connection to NNMi console, 73 to 76

seeds, rule-based discovery, 47

self-signed certificates
installing public key certificates, 74
updating server.xml, 75

sendMsg.ovpl command, NNM 6.x/7.x, 181 to 182

server.xml file, 74 to 76

servers
AlarmPoint, 223

Service Level Agreements, 63

services
NmsApa
configuration changes, 268
device-generated traps, 267
network connections, 267
node unreachable, 263
node updated, 267
status, 242

settings
communication, 42
configuration, 33
meaningful defaults for communication, 39
monitoring, 67
object status, 242
verifying default, 66

shadow
clearing, 262
creating, 261

shared configuration data, 126

shared disk
configuring, 97
copying data files, 84
data, 96
files, 117
support, 78
unmounting, 106, 108

shared file system type, HA configuration, 85, 92

shared HA data, 96 to 97

short name, virtual host for HA configuration, 84, 92

SLAs, 63

SNMP
agent queries
responsive, 248
unresponsive, 247
agent status, 244
communication problems, 54
community strings, 36
Component Health, 64
configuring access, 148 to 152
Data Presenter, 180
device access, 42
disabling
communication, 40
traffic, 38
MIB directories, 127
migrating to NNMi, 148 to 152
monitoring, 64
node configuration, 42
protocol, 35, 243
requests, 43
tuning settings, 43
version preferences, 36

snmpout.txt file, 149
snmptrap command, NNM 6.x/7.x, 182
software
  requirements, 19
  supported, 21
Solaris
  configuring shared disks, 97
  NIS+, 100
  nnmharg.ovpl script, 119
  nnmhargconfigure.ovpl
    command, 114
    script, 119
  shared disk format, 85, 93
  Veritas Cluster Server, 78
  virtual hosts, 85, 92
Spiral Discovery, 45
ssh command, 78
SSL technology, 73
starting
  HA resource group, 114
  NNMi after HA maintenance, 102
  NNM iSPI for Performance after HA
    maintenance, 103
startup problems
  nmsdbmgr, 116
  NNMi, 115
  pmd, 116
State Poller
  concepts, 57 to 58
  configuring, 64 to 66
  evaluating configuration, 66 to 68
  ICMP, 38
  planning, 58 to 64
  symptoms, 242
  tuning, 68 to 69
  verifying
    communication settings, 43
    health statistics, 68
status
  incident category, 178
  interface, 244
  monitoring
    concepts, 141
    configuring, 139
    migrating from NNM 6.x/7.x, 161 to 166
  node groups, 245
  nodes, 244
  object, 242
  poll
    evaluating performance, 67 to 68
    initiating, 67
  SNMP agent, 244
Status Alarms category, NNM 6.x/7.x, 178
Status incident category, 178
stopping
  HA resource group
    NNMi, 105
    NNM iSPI for Performance, 108
  NNM iSPI for Performance without causing HA
    failover, 103
  NNMi without causing HA failover, 102
strategy, backup, 123 to 124
strings, community
  access credentials, 36
  multiple, 41
supported
  HA products, 78
  hardware and software, 21
supported versions
  AlarmPoint integration, 222
  HPOM integration, 189
  HP UCMDB integration, 214
  nGenius Performance Manager integration, 226
sweep, 49
switches
  defaults, 56
  defining node groups, 30
  discovering, 55
  hierarchies, 31
synchronization, HPOM, 197
synchronization, troubleshooting HPOM, 197
syntax, backup script, 125
syslog.log file, 120
system
  device support matrix, 21
  object ID ranges
    auto-discovery, 53
    evaluating, 56
  resources, 69
  shared file type, HA configuration, 85, 92

## T

tables, configuration, 126
tar files, 126, 128
task flow model, 27
terms
  HA, 80
test environment, 15

testing
communication configuration, 41
lag time, 40
NNM 6.x/7.x
integration, 180
traps, 182

third-party certificates, 74

Threshold Alarms category, NNM 6.x/7.x, 178

timeouts, 36
configuring, 37
tuning, 43
values, 40

tips, configuration
auto-discovery, 53
seeded discovery, 53

Tomcat configuration, 73

topology
database, 57
NNM 6.x/7.x management station, 176
scope data, 127
summary example URL, 180

traceroute command, 40

traffic
controlling, 40
disabling, 38

trapd.conf file, editing, 176

traps
generating, 267
testing to NNM 6.x/7.x system, 182

troubleshooting
AlarmPoint integration, 224
discovery
auto-discovery, 55 to 56
HA configuration, 113 to 117
HPOM integration, 195 to 197
firewall, 197
synchronization, 197
nGenius Performance Manager integration, 226 to 228

tuning
communication, 43
state polling, 68 to 69

## U

unconfiguring HA clusters
NNMi, 104 to 107
NNM iSPI for Performance, 107 to 109

underlying cause, 242

UNIX
disk group, 85, 93
environment variables
administration, 237
installation, 16
HA configuration
files, 118
log files, 120
scripts, 118
HPOM integration
enabling, 191
versions, 189
ipNoLookup.conf command, 151
jboss command, 74
LicFile.txt file
configuring primary HA cluster node, 86
installing HA cluster license, 98
mount point, 85, 93
netmon.cmstr command, 148
nGeniusNNM8.zip file, 227
nnm8EventForwardDestinations.txt file, 175
nnm.envvars.sh command, 236
nnmlicense.ovpl command, 98
nnmtrapd log files, 116
pmd log files, 116
port numbers, 179
remsh command, 78
sendMsg.ovpl command, 182
shared disk directories, 97
ssh command, 78
trapd.conf file, 176
volume group, 85, 93

unmounting shared disk, 106, 108

unreachable nodes, 263

unresponsive
IPv4Address to ICMP, 249
SNMP agent to queries, 247

up
connection
operationally, 256
router, 264
interface
administratively, 254
operationally, 252
node
clearing shadow, 262
distribution router, 258
router, 264

updating
nodes, 267
server.xml, 74 to 76

URLs
not requiring selection, 180
requiring selection, 180

user interface model, 28

users, communicating https connection information, 76

UUID, node details using, 180

## V

variables, environment
- overview, 235
- UNIX
  - administration, 237
  - installation, 16
  - MANPATH, 22
- Windows
  - administration, 236
  - installation, 16
  - NNMi 8.00, 238

variables, MIB II, 64

verifying
- communication settings, 42
- default settings, 66
- event forwarding, 181
- interface groups, 66
- management IP address, 43
- monitoring configuration, 66 to 67
- network
  - connectivity, 54
  - monitoring configuration, 66
- NNM 6.x/7.x incident configuration, 177
- node groups, 66
- nodes configured for SNMP, 42
- order numbers, 65
- SNMP access for devices, 42
- State Poller
  - communication settings, 43
  - health statistics, 68

Veritas Cluster Server, 78
- HA resource group, 80
- nnmharg.ovpl script, 119

version
- information file, 126
- SNMP preferences, 36

version comparisons
- customizing
  - event monitoring, 142
- discovering networks, 136
- status monitoring, 139

versions, HPOM integration, 189

versions, supported
- AlarmPoint, 222
- HPOM, 189
- HP UCMDB, 214
- nGenius Performance Manager, 226

views
- configuring additional NNM 6.x/7.x, 179
- launching dynamic from NNMi, 183

virtual host, HA configuration
- netmask, 84, 92
- network interface, 85, 92
- short name, 84, 92

virtual IP address support, 78

Volume_A.log file, 121

volume group, HA configuration, 85, 93

vxfs shared disk format, 85, 93

## W

web server
- AlarmPoint, 223
- enabling https protocol, 73
- ports, 179
- public key certificates, 73

weekly data backups, 123

window, About HP Network Node Manager i-series, 68

Windows
commands
nnmhargconfigure.ovpl, 114
configuring shared disks, 97
environment variables
administration, 236 to 238
installation, 16
HA configuration
files, 118
log files, 120
scripts, 118
HPOM integration
enabling, 190
versions, 189
ipNoLookup.conf command, 151
jboss command, 74
LicFile.txt file
configuring primary HA cluster node, 86
installing HA cluster license, 98
Microsoft Cluster Services, 78
mount point, 85, 93
netmon.cmstr command, 148
nGeniusNNM8.zip file, 227
nnm8EventForwardDestinations.txt file, 175
nnm.envvars.bat command, 236 to 238
nnmlicense.ovpl command, 98
nnmtrapd log files, 116
pmd log files, 116
port numbers, 179
sendMsg.ovpl commands, 182
shared disk directories, 96
shared disk format, 85, 93
trapd.conf file, 176
virtual hosts, 85, 92

wizard, NNMi Quick Start Configuration, 15

workspaces
Configuration
configuring NNM 6.x/7.x management station, 179
configuring state polling, 64
evaluating state polling, 66
Incidents, 181
Interface Groups, 61
Node Groups, 61

## X

XML files, 34, 132

xnmevents command, NNM 6.x/7.x, 175

XServer, 176

hp
invent

Java™
COMPATIBLE