

JP1 Version 12

JP1/Network Node Manager i Setup Guide

3021-3-E02-20(E)

Notices

Relevant program products

For Windows Server 2012, Windows Server 2012 R2, Windows Server 2016, and Windows Server 2019

P-2942-82CL JP1/Network Node Manager i 12-50

P-2942-89CL JP1/Network Node Manager i Developer's Toolkit 12-00

For CentOS 6.1 (x64) and later, CentOS 7.1 and later, CentOS 8.1 and later, Linux 6.1 (x64) and later, Linux 7.1 and later, Linux 8.1 and later, Oracle Linux 6.1 and later, Oracle Linux 7.1 and later, Oracle Linux 8.1 and later, and SUSE Linux 12

P-8242-82CL JP1/Network Node Manager i 12-50 P-8242-89CL JP1/Network Node Manager i Developer's Toolkit 12-00

Trademarks

HITACHI, HA Monitor, Job Management Partner 1, and JP1 are either trademarks or registered trademarks of Hitachi, Ltd. in Japan and other countries.

Active Directory is either a registered trademark or trademark of Microsoft Corporation in the United States and/or other countries.

AIX is a trademark of International Business Machines Corporation, registered in many jurisdictions worldwide. Cisco is a registered trademark or trademark of Cisco Systems, Inc. and/or its affiliates in the United States and certain other countries.

Cisco ACI is a registered trademark or trademark of Cisco Systems, Inc. and/or its affiliates in the United States and certain other countries.

Excel is either a registered trademark or trademark of Microsoft Corporation in the United States and/or other countries. Internet Explorer is either a registered trademark or trademark of Microsoft Corporation in the United States and/or other countries.

Itanium is a trademark of Intel Corporation or its subsidiaries in the U.S. and/or other countries.

Linux is the registered trademark of Linus Torvalds in the U.S. and other countries.

Microsoft is either a registered trademark or trademark of Microsoft Corporation in the United States and/or other countries.

Microsoft Edge is either a registered trademark or trademark of Microsoft Corporation in the United States and/or other countries.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

PostgreSQL is a registered trademark or trademark of the PostgreSQL Community Association of Canada in Canada and/or other countries.

Red Hat is a registered trademark of Red Hat, Inc. in the United States and other countries.

Red Hat Enterprise Linux is a registered trademark of Red Hat, Inc. in the United States and other countries.

Symantec, Symantec are trademarks or registered trademarks of Symantec Corporation or its affiliates in the U.S. and other countries.

Veritas, the Veritas Logo, and VERITAS are trademarks or registered trademarks of Veritas Technologies LLC or its affiliates in the U.S. and other countries. Other names may be trademarks of their respective owners.

VMware is a registered trademark or trademark of VMware, Inc. in the United States and other jurisdictions.

Windows is either a registered trademark or trademark of Microsoft Corporation in the United States and/or other countries.

Windows Server is either a registered trademark or trademark of Microsoft Corporation in the United States and/or other countries.

Other company and product names mentioned in this document may be the trademarks of their respective owners.

Microsoft product name abbreviations

This manual uses the following abbreviations for Microsoft product names.

Abbreviation		Full name or meaning	
Active Directory		Microsoft ^(R) Active Directory	
Excel		Microsoft ^(R) Office Excel	
Internet Explorer		Windows ^(R) Internet Explorer ^(R)	
Microsoft Cluster Service		Microsoft ^(R) Cluster Service	
Windows	Windows Server 2012	Windows Server 2012	Microsoft ^(R) Windows Server ^(R) 2012 Datacenter
			Microsoft ^(R) Windows Server ^(R) 2012 Standard
		Windows Server 2012 R2	Microsoft ^(R) Windows Server ^(R) 2012 R2 Datacenter
			Microsoft ^(R) Windows Server ^(R) 2012 R2 Standard
	Windows Server 2016		Microsoft ^(R) Windows Server ^(R) 2016 Datacenter
			Microsoft ^(R) Windows Server ^(R) 2016 Standard
	Windows Server 2019		Microsoft ^(R) Windows Server ^(R) 2019 Datacenter
			Microsoft ^(R) Windows Server ^(R) 2019 Standard
WSFC		Microsoft ^(R) Windows Server ^(R) Failover Cluster	

Acknowledgements

This product includes software developed by the Apache Software Foundation.

(http://www.apache.org)

This product includes software developed by the Indiana University Extreme! Lab.

(http://www.extreme.indiana.edu)

This product includes software developed by The Legion Of The Bouncy Castle.

(http://www.bouncycastle.org)

Issued

Mar. 2021: 3021-3-E02-20(E)

Copyright

© Copyright 2009-2021 Micro Focus or one of its affiliates.

All Rights Reserved. Copyright (C) 2019, 2021, Hitachi, Ltd.

This software and documentation are based in part on software and documentation under license from Micro Focus or one of its affiliates.

Summary of amendments

The following table lists changes in this manual (3021-3-E02-20(E)) and product changes related to this manual.

Changes	Location
A description about kernel.shmall was added to the preinstallation checklist for the NNMi management server.	Table 1-1
A description about the specification of HTTPS port numbers during the installation of NNMi was added.	2.1.1, 2.1.2
LANG was added as an environment variable to be specified during the installation of NNMi.	2.1.2
A description about supported certificate formats was added.	10.3.2, 10.4.3(1)
The procedure for replacing a certificate that was created during the installation of NNMi was changed.	10.3.4
A description about cases where a root CA certificate needs to be imported for multiple LDAP directory servers was added.	10.3.8
Microsoft Edge was added as a browser to be configured for a telnet or SSH client.	11., 11.2
The description about Windows operating systems that run Internet Explorer or Firefox was changed.	11.2
A description about server elements was changed.	12.2.3
A description about the default port numbers set by the NNMi installation script was added.	15.4.2
The expression for enabling the function that auto-trims the oldest incidents was changed.	21.20.1
The expression for enabling the function that auto-trims the oldest SNMP trap incidents was changed.	21.20.2
A description about the rotation of archive files was added.	21.20.3
A description in <i>Upgrading from NNMi Version 9, 10, or 11</i> , was changed to explain how to upgrade to version 12-50.	24.
In the list of ports used by the NNMi management server, a description about changing the settings of port 443 was added.	Table E-1

In addition to the above changes, minor editorial corrections were made.

Preface

This manual describes the settings needed to deploy *JP1/Network Node Manager i* and *JP1/Network Node Manager i Advanced* (referred to hereafter as *NNMi* if there is no difference in the products). It also describes the settings needed to upgrade from version 8 and earlier versions of JP1/Cm2/Network Node Manager (abbreviated as *NNM*). JP1/Cm2/ Network Node Manager is a product name in Japan.

Note also that this manual is intended for all supported operating systems. When there are differences between the NNMi editions on different operating systems, this manual provides separate descriptions for the relevant operating systems.

Intended readers

This manual is intended for users who evaluate and implement a network distributed management system configuration that uses NNMi. This manual assumes that the readers are experienced system administrators, network engineers, and others who are familiar with deploying and managing networks for large-scale systems.

Organization of this manual

This manual is organized into the following parts:

PART 1: Preparation

Part 1 explains the preparations required before NNMi is installed, as well as the procedures for installation and uninstallation of NNMi.

PART 2: Introduction

Part 2 explains the minimum settings required for starting network management when NNMi is used.

PART 3: Configuration

Part 3 explains the settings needed for managing a network.

PART 4: Advanced Configuration

Part 4 explains the settings needed to use NNMi functions, such as certification, integration of directory services through NNMi and LDAP, and so on.

PART 5: High Availability Environment Configuration

Part 5 describes support for high availability (HA) clusters and application failover.

PART 6: NNMi Maintenance

Part 6 explains how to back up, restore, and maintain NNMi.

PART 7: Migration

Part 7 explains the operations needed to upgrade NNMi to version 12.

PART 8: Integration with NNMi

Part 8 explains integration of related products with NNMi.

PART 9: Integration

Part 9 explains linkage of NNMi with related products.

Correspondence between previous and current edition manuals

The manual listed below have been incorporated into this manual, *JP1/Network Node Manager i Setup Guide* (3021-3-E02(E)), which has been updated with the enhancements of version 12:

 Job Management Partner 1/Consolidated Management 2/Network Node Manager i Installation Guide (3021-3-342-20 (E))

The table below shows the correspondence between the previous and current edition setup guides.

Correspondence between the previous and current edition setup guides

Previous Edition Setup Guide (3021-3-343-20(E))	Current Edition Setup Guide (3021-3-E02(E))
PART 1: Preparation	PART 1: Preparation
1 Hardware and Software Requirements	
1.1 Supported hardware and software	(Corresponds to 1.1 Checking the hardware and software)
1.2 System configuration (UNIX)	(This section was moved to Appendix A. When NNMi Manpages Cannot Be Displayed (Linux).)
(2. Preinstallation Checklists was inserted from the Installation Guide into the current	1. Preinstallation Checklists
edition Setup Guide)	1.1 Checking the hardware and software
	1.2 Preparing the preinstallation NNMi management server environment
	1.3 Checking for a well-configured DNS
	1.4 Preparing to use the NNMi Quick Start Configuration Wizard
(A. Additional Information About Installation was inserted from the Installation Guide into the current edition Setup Guide)	1.5 Additional information about installation
(3. Installing and Uninstalling NNMi was inserted from the Installation Guide into	2 Installing and Uninstalling NNMi
the current edition Setup Guide)	2.1 Installing NNMi
	2.2 Using the Quick Start Configuration Wizard
	2.3 Licensing NNMi
	2.4 Removing NNMi
(<i>B. Troubleshooting Installation and Initial Startup</i> was inserted from the <i>Installation Guide</i> into the current edition Setup Guide)	2.5 Troubleshooting installation and initial startup
(4. Getting Started with NNMi was inserted from the Installation Guide into the current	Part 2: Introduction
edition Setup Guide)	3. Getting Started with NNMi
	3.1 Accessing NNMi
	3.2 Accessing NNMi Help
	3.3 Configuring network discovery
PART 2: Configuration	PART 3: Configuration

Previous Edition Setup Guide (3021-3-343-20(E))	Current Edition Setup Guide (3021-3-E02(E))
2. General Concepts for Configuration	4. General Concepts for Configuration
3. NNMi Communication	5. NNMi Communications
4. NNMi Discovery	6. NNMi Discovery
5. NNMi State Polling	7. NNMi State Polling
6. NNMi Incidents	8. NNMi Incidents
7. NNMi Console	9. NNMi Console
PART 3: Advanced Configuration	PART 4: Advanced Configuration
8. Working with Certificates for NNMi	10. Working with Certificates for NNMi
9. Configuring the Telnet and SSH Protocols for Use by NNMi	11. Configuring the Telnet and SSH Protocols for Use by NNMi
10. Integrating NNMi with a Directory Service Through LDAP	12. Integrating NNMi with a Directory Service Through LDAP
11. Managing Overlapping IP Addresses in a NAT Environment	13. Managing Overlapping IP Addresses in a NAT Environment
12. NNMi Security and Multi-Tenancy	14. NNMi Security and Multi-Tenancy
13. Global Network Management	15. Global Network Management
14. NNMi IPv6 Management Feature	16. NNMi IPv6 Management Feature
PART 4: High Availability Environment Configuration	PART 5: High Availability Environment Configuration
15. NNMi Data Resilience	17. NNMi Data Resilience
16. Configuring NNMi for Application Failover	18. Configuring NNMi for Application Failover
17. Configuring NNMi in a High Availability Cluster	19. Configuring NNMi in a High Availability Cluster
PART 5: NNMi Maintenance	PART 6: NNMi Maintenance
18. NNMi Backup and Restore Tools	20. NNMi Backup and Restore Tools
19. Maintaining NNMi	21. Maintaining NNMi
20. Changing the NNMi Management Server	22. Changing the NNMi Management Server
21. NNMi Security	23. NNMi Security
PART 6: Migration	PART 7: Migration
22. Upgrading from NNMi Version 9, 10-00, or 10-10	24. Upgrading from NNMi Version 9, 10, or 11
23. Comparison with NNM Version 8 or Earlier	25. Comparison with NNM Version 8 or Earlier
24. Upgrading from NNM Version 8 or Earlier	26. Upgrading from NNM Version 8 or Earlier
	27. Migrating NNMi from an HP-UX or Solaris Operating System
PART 7: Integration with NNMi	PART 8: Integration with NNMi
25. NNMi Northbound Interface	28. NNMi Northbound Interface
	PART 9: Integration

Previous Edition Setup Guide (3021-3-343-20(E))	Current Edition Setup Guide (3021-3-E02(E))
	29. Integration with JP1/IM2 Intelligent Integrated Management Base
	30. RESTful API
Appendixes	Appendixes
(This appendix was moved from 1.2 System configuration (UNIX).)	A. When NNMi Manpages Cannot Be Displayed (Linux)
(C. List of MIBs Read During a New Installation was inserted from the Installation Guide into the current edition Setup Guide)	B. List of MIBs Read During a New Installation
A. NNMi Environment Variables	C. NNMi Environment Variables
B. The Causal Engine and NNMi Incidents	D. The Causal Engine and NNMi Incidents
C. List of Ports Used by NNMi	E. List of Ports Used by NNMi
D. Version Changes	F. Version Changes
E. Reference Material for This Manual	G. Reference Material for This Manual
F. Glossary	H. Glossary

Conventions: Fonts and symbols

The following table explains the text formatting conventions used in this manual:

Text formatting	Convention
Bold	 Bold characters indicate text in a window, other than the window title. Such text includes menus, menu options, buttons, radio box options, or explanatory labels. For example: From the File menu, choose Open. Click the Cancel button. In the Enter name entry box, type your name.
Italic	 Italic characters indicate a placeholder for some actual text to be provided by the user or system. For example: Write the command as follows: copy <i>source-file target-file</i> The following message appears: A file was not found. (file = <i>file-name</i>) Italic characters are also used for emphasis. For example: Do <i>not</i> delete the configuration file.
Monospace	 Monospace characters indicate text that the user enters without change, or text (such as messages) output by the system. For example: At the prompt, enter dir. Use the send command to send mail. The following message is displayed: The password is incorrect.

The following table explains the symbols used in this manual:

Symbol	Convention
1	In syntax explanations, a vertical bar separates multiple items, and has the meaning of OR. For example: A B C means A, or B, or C.
{ }	In syntax explanations, curly brackets indicate that only one of the enclosed items is to be selected. For example: $\{A B C\}$ means only one of A, or B, or C.
[]	In syntax explanations, square brackets indicate that the enclosed item or items are optional. For example: [A] means that you can specify A or nothing. [B C] means that you can specify B, or C, or nothing.
	In coding, an ellipsis () indicates that one or more lines of coding have been omitted. In syntax explanations, an ellipsis indicates that the immediately preceding item can be repeated as many times as necessary. For example: A, B, B, means that, after you specify A, B, you can specify B as many times as necessary.

Contents

Notices2Summary of amendments4Preface5

Part 1: Preparation

1	Preinstallation Checklists 26
1.1	Checking the hardware and software 27
1.2	Preparing the preinstallation NNMi management server environment 28
1.3	Checking for a well-configured DNS 32
1.4	Preparing to use the NNMi Quick Start Configuration Wizard 34
1.5	Additional information about installation 35
1.5.1	Specifying disk drive security settings (Windows) 35
1.5.2	Obtaining or setting the official fully qualified domain name 35
1.5.3	Enabling the Web browser for the NNMi console 35
1.5.4	Installing required libraries in Linux 37
1.5.5	Setting the system account password 38
2	Installing and Uninstalling NNMi 39
2.1	Installing NNMi 40
2.1.1	Installing NNMi (Windows) 40
2.1.2	Installing NNMi (Linux) 43
2.1.3	Operations after the installer finishes 44
2.2	Using the Quick Start Configuration Wizard 48
2.3	Licensing NNMi 52
2.3.1	Preparing to install a permanent license key 52
2.3.2	Obtaining and installing a permanent license key 52
2.3.3	Switching the temporary license 52
2.4	Removing NNMi 54
2.4.1	Removing NNMi (Windows) 54
2.4.2	Removing NNMi (Linux) 55
2.5	Troubleshooting installation and initial startup 57
2.5.1	Installation problems 57
2.5.2	Initial startup problems 58

Part 2: Introduction

3 Getting Started with NNMi 61

- 3.1 Accessing NNMi 62
- 3.2 Accessing NNMi Help 63
- 3.3 Configuring network discovery 64
- 3.3.1 Configuring community strings 64
- 3.3.2 Configuring auto-discovery rules 65
- 3.3.3 Checking discovery progress 67

Part 3: Configuration

4 General Concepts for Configuration 68

- 4.1 Task flow model 69
- 4.2 Best practice: Save the existing configuration 70
- 4.3 Best practice: Use the Author attribute 71
- 4.4 User interface model 72
- 4.5 Ordering 73
- 4.6 Node groups and interface groups 74
- 4.6.1 Group overlap 74
- 4.6.2 Node group membership 75
- 4.6.3 Node group status 77
- 4.6.4 Interface groups 77
- 4.7 Node/interface/address hierarchy 79
- 4.8 Resetting the NNMi configuration and database 80

5 NNMi Communications 81

- 5.1 Concepts for communications 82
- 5.1.1 Levels of communication configuration 82
- 5.1.2 Network latency and timeouts 83
- 5.1.3 SNMP access control 83
- 5.1.4 SNMP version preferences 84
- 5.1.5 Management address preferences 85
- 5.1.6 SNMPv3 traps and informs 85
- 5.1.7 Polling protocols 86
- 5.1.8 nnmsnmp*.ovpl commands 86
- 5.2 Creating a communication plan 87
- 5.2.1 Default communication settings 87
- 5.2.2 Communication configuration regions 87
- 5.2.3 Specific node configurations 88
- 5.2.4 Retry and timeout values 88
- 5.2.5 Active protocols 88

5.2.6 Community strings and authentication profiles 89 Configuring communications 5.3 91 5.3.1 Configuring an SNMP proxy 91 5.3.2 Using NETCONF for device support 92 5.3.3 Discover and monitor VMware hypervisor-based virtual networks 94 Discover and monitor Cisco ACI networks 97 5.3.4 5.3.5 Multihomed NNMi Management Server 99 5.4 Evaluating communications 101 5.4.1 Checking all nodes for SNMP configuration 101 Checking SNMP access 101 5.4.2 5.4.3 Checking the management IP address for SNMP Devices 101 5.4.4 Checking the communication settings 102 5.4.5 Checking whether the monitoring configuration matches the communication settings 102 5.5 Tuning communications 103

6 NNMi Discovery 104

- 6.1 Concepts of discovery 105
- 6.1.1 Device profiles and device attributes 106
- 6.2 Planning discovery 107
- 6.2.1 Selecting your primary discovery approach 107
- 6.2.2 Creating auto-discovery rules 108
- 6.2.3 Changing the order for node name resolution 110
- 6.2.4 Subnet connection rules 111
- 6.2.5 Discovery seeds 111
- 6.2.6 Rediscovery interval 112
- 6.2.7 Do-not-discover objects 112
- 6.2.8 Interface discovery range 113
- 6.3 Configuring discovery 114
- 6.3.1 Tips for configuring auto-discovery rules 114
- 6.3.2 Tips for configuring seeds 114
- 6.3.3 Discovering link aggregation 115
- 6.3.4 Discovering server-to-switch link aggregation (S2SLA) 115
- 6.4 Evaluating discovery 117
- 6.4.1 Following the progress of initial discovery 117
- 6.4.2 Checking for discovery of all seeds 117
- 6.4.3 Checking for valid device profiles 118
- 6.4.4 Checking for discovery of all nodes 118
- 6.4.5 Evaluating the auto-discovery rules (Rule-based discovery only) 118
- 6.4.6 Evaluating connections and VLANs 119
- 6.4.7 Rediscovering a device 119
- 6.5 Tuning discovery 120
- 6.5.1 Deleting unresponsive objects 120

7	NNMi State Polling 121
7.1	Concepts for state polling 122
7.1.1	Order of evaluation 122
7.2	Planning state polling 123
7.2.1	Polling checklist 123
7.2.2	What Can NNMi Monitor? 124
7.2.3	Stop Monitoring 125
7.2.4	Interfaces to Unmonitored Nodes 125
7.2.5	Extending Monitoring 125
7.2.6	Creating node and interface groups 126
7.2.7	Planning polling intervals 128
7.2.8	Planning the data to be collected 129
7.2.9	Deciding which SNMP traps to send to NNMi 129
7.3	Configuring state polling 132
7.3.1	Configuring the interface groups and node groups to be monitored 132
7.3.2	Configuring interface monitoring 132
7.3.3	Configuring node monitoring 133
7.3.4	Specifying the default settings for monitoring 133
7.4	Evaluating state polling 135
7.4.1	Verifying the configuration for network monitoring 135
7.4.2	Evaluating the performance of status polling 136
7.5	Tuning state polling 138
8	NNMi Incidents 139
8.1	Concepts for incidents 140
8.1.1	Incident lifecycle 140
8.1.2	Trap and incident forwarding 141
8.1.3	Received SNMP traps 142
8.1.4	MIBs 143
8.1.5	Custom incident attributes 143
8.1.6	Incident reduction 144
8.1.7	Incident suppression, enrichment, and dampening 144
8.1.8	Lifecycle transition actions 145
8.2	Planning incidents 147
8.2.1	Planning the SNMP traps to be processed 147
8.2.2	Planning the incidents to be displayed 147
8.2.3	Planning how NNMi responds to incidents 147
8.3	Configuring incidents 148
8.3.1	Configuring incident suppression, enrichment, and dampening 148

- 8.3.2 Configuring lifecycle transition actions 148
- 8.3.3 Configuring trap logs 149

- 8.3.4 Configuring incident logs 149
- 8.3.5 Configuring trap server properties 149
- 8.4 Batch loading incident configuration 151
- 8.4.1 Using nnmincidentcfgdump.ovpl to create an incident configuration file 151
- 8.4.2 Using nnmincidentcfgload.ovpl to load the incident configuration 151
- 8.5 Evaluating incidents 153
- 8.6 Tuning incidents 154
- 8.6.1 Enabling incidents for undefined traps 154
- 8.6.2 Interpreting and displaying the MIB data for SNMP traps correctly 155

9 NNMi Console 156

- 9.1 A practical example of using node groups 157
- 9.1.1 Creating node groups 157
- 9.1.2 Configuring the node group maps 160
- 9.1.3 Deleting node groups 162
- 9.2 Reducing the maximum number of nodes displayed in a Network Overview map 163
- 9.3 Reducing the number of displayed nodes on a node group map 164
- 9.4 Configuring Gauges in the Analysis Pane 165
- 9.4.1 Disabling the Analysis pane 165
- 9.4.2 Limiting the Number of Gauges Displayed 166
- 9.4.3 Setting the Refresh Rate for Gauges in the Analysis Pane 166
- 9.4.4 Eliminating Gauges from the Display 166
- 9.4.5 Controlling the Order of Displayed Node Gauges 166
- 9.4.6 Controlling the Order of Displayed Interface Gauges 167
- 9.4.7 Controlling the Order of Displayed Custom Poller Gauges 167
- 9.4.8 Understanding how Gauge Properties are Applied 167
- 9.4.9 Troubleshooting Gauge Problems 168
- 9.5 Configuring Map Label Scale Size and Borders 169
- 9.6 Configuring Auto-Collapse Thresholds for Loom and Wheel Diagrams 170
- 9.7 Customizing device profile icons 171
- 9.8 Overriding the refresh rate of table views 172

Part 4: Advanced Configuration

- 10 Working with Certificates for NNMi 173
- 10.1 About NNMi Certificates 174
- 10.2 Configuring an Upgraded NNMi Environment to Use the New Keystore 176
- 10.3 Using Certificates with the PKCS #12 Repository 179
- 10.3.1 Generating a Self-Signed Certificate 179
- 10.3.2 Generating a CA-Signed Certificate 180
- 10.3.3 Delete a Certificate from the NNMi Keystore 185
- 10.3.4 Replacing an Existing Certificate with a new Self-Signed or CA-Signed Certificate 186

- 10.3.5 Working with Certificates in Application Failover Environments 187
- 10.3.6 Working with Certificates in High-Availability Environments 188
- 10.3.7 Working with Certificates in Global Network Management Environments 189
- 10.3.8 Configuring an SSL connection to the Directory service 191
- 10.4 Using Certificates with the JKS Repository 193
- 10.4.1 Replacing an Existing Certificate with a new Self-Signed or CA-Signed Certificate 194
- 10.4.2 Generating a Self-Signed Certificate 194
- 10.4.3 Generating a CA-Signed Certificate 195
- 10.4.4 Configuring application failover to use self-signed certificates 200
- 10.4.5 Working with Certificates in High-Availability Environments 201
- 10.4.6 Working with Certificates in Global Network Management Environments 202
- 10.4.7 Configuring an SSL connection to the directory service 203

11 Configuring the Telnet and SSH Protocols for Use by NNMi 206

- 11.1 Disable the telnet or SSH menu item 207
- 11.2 Configure a telnet or SSH client for the browser on Windows 208
- 11.2.1 Windows operating system-provided telnet client 209
- 11.2.2 Third-party telnet client (standard Windows) 210
- 11.2.3 Third-party telnet client (Windows on Windows) 211
- 11.2.4 Third-party SSH client (standard Windows and Windows on Windows) 212
- 11.3 Configure Firefox to use telnet or SSH on Linux 214
- 11.3.1 Configuring Firefox to use Telnet on Linux 214
- 11.3.2 Configuring Firefox to use SSH on Linux 214
- 11.4 Example files for changing the Windows registry 216
- 11.4.1 Example nnmtelnet.reg 216
- 11.4.2 Example nnmputtytelnet.reg 216
- 11.4.3 Example nnmtelnet32on64.reg 216
- 11.4.4 Example nnmssh.reg 216

12 Integrating NNMi with a Directory Service Through LDAP 218

- 12.1 NNMi user access information and configuration options 219
- 12.1.1 Internal mode: Storing all NNMi user information in the NNMi database 220
- 12.1.2 Mixed mode: Storing some NNMi user information in the NNMi database and some NNMi user information in a directory service 221
- 12.1.3 External mode: Storing all NNMi user information in a directory service 222
- 12.2 Configuring NNMi to access a directory service 224
- 12.2.1 Task 1: Back up the current NNMi user information 225
- 12.2.2 Task 2: (Optional) Configure secure communications to the directory service 225
- 12.2.3 Task 3: Configure user access from the directory service 225
- 12.2.4 Task 4: Test the user name and password configuration 227
- 12.2.5 Task 5: (Configuring for the external mode only) Configure group retrieval from the directory service 228

- 12.2.6 Task 6: (Configuring for the external mode only) Map the directory service groups to NNMi user groups 229
- 12.2.7 Task 7: (Configuring for the external mode only) Test the NNMi user group configuration 230
- 12.2.8 Task 8: (Configuring for the external mode only) Configure NNMi user groups for incident assignment 230
- 12.2.9 Task 9: Clean up to prevent unexpected access to NNMi 231
- 12.2.10 Task 10: (Optional) Map the user groups to security groups 231
- 12.3 Directory service queries 232
- 12.3.1 Directory service access 232
- 12.3.2 Directory service content 232
- 12.3.3 Information owned by the directory service administrator 235
- 12.3.4 User identification 236
- 12.3.5 User group identification 237
- 12.4 Directory service configuration for storing NNMi user groups 240
- 12.5 Troubleshooting the directory service integration 241
- 12.6 LDAP configuration file reference 242
- 12.6.1 nms-auth-config.xml 242
- 12.7 Switching to the nms-auth-config.xml File 244

13 Managing Overlapping IP Addresses in a NAT Environment 245

- 13.1 About NAT 246
- 13.2Benefits of NAT247
- 13.3 Supported NAT types 248
- 13.4 How to implement NAT in NNMi 249
- 13.5 Considerations on static NAT 250
- 13.5.1 Hardware and software requirements for static NAT 251
- 13.5.2 Communication using static NAT 251
- 13.5.3 Discovery and static NAT 252
- 13.5.4 Monitoring configuration for static NAT 253
- 13.5.5 Traps and static NAT 253
- 13.5.6 Subnets and static NAT 258
- 13.5.7 Global network management and static NAT 258
- 13.6 Dynamic NAT and dynamic PAT considerations 259
- 13.6.1 Hardware and software requirements for dynamic NAT and dynamic PAT 261
- 13.6.2 Discovery and dynamic NAT or dynamic PAT 261
- 13.6.3Monitoring configuration for dynamic NAT261
- 13.6.4 Subnets and dynamic NAT or dynamic PAT 262
- 13.6.5 Global network management and dynamic NAT or dynamic PAT 262
- 13.6.6 Deploying NNMi in a network address translation (NAT) environment 262
- 13.6.7 NNMi calculations for state and status 264
- 13.7 Mapping overlapping IP addresses 266
- 13.7.1 Ranges of private IP addresses 266

14	NNMi Security and Multi-Tenancy 267
14.1	Effects of limiting object access 268
14.2	The NNMi security model 269
14.2.1	Security groups 269
14.2.2	Example security group structure 270
14.3	The NNMi tenant model 273
14.3.1	Tenants 273
14.3.2	Example tenant structure 274
14.4	NNMi security and multi-tenancy configuration 276
14.4.1	Security and multi-tenancy configuration tools 276
14.4.2	Configuring multi-tenancy 278
14.4.3	Configuring security groups 279
14.4.4	Verifying the configuration 281
14.4.5	Exporting the NNMi security and multi-tenancy configuration 282
14.5	Defining NNMi security and multi-tenancy in global network management 283
14.5.1	Initial configuration of security and multi-tenancy in global network management 283
14.5.2	Effects of security and multi-tenancy assignment on a global network management 284
15	Global Network Management 286
15.1	Prerequisites for global network management 287
15.2	Global network management benefits 288
15.3	Evaluating the use of global network management 289
15.3.1	Monitoring networks at multiple sites continuously 289
15.3.2	Monitoring selected critical devices 289
15.3.3	Considering licensing 289
15.4	Practical global network management examples 290
15.4.1	Reviewing the requirements 290
15.4.2	Initial preparation 291
15.5	Configuring forwarding filters on the regional managers 295
15.5.1	Configuring a forwarding filter to limit forwarded nodes 295
15.6	Connecting a global manager with a regional manager 307
15.7	Determining the connection status from global1 to regional1 and regional2 311
15.8	Reviewing global1 inventory 313
15.9	Disconnecting communication between global1 and regional1 316
15.10	Additional information about global network management 320
15.10.1	Discovery and data synchronization 320
15.10.2	Replicating custom attributes from a regional manager to the global manager 322
15.10.3	Status polling or configuration polling a device 322
15.10.4	Determining device status and NNMi incident generation using a global manager 324
15.11	Troubleshooting tips for global network management 325
15.11.1	Troubleshooting information in NNMi Help 325

15.11.2 Clock synchronization 325 Global network management system information 15.11.3 325 15.11.4 Synchronizing regional manager discovery from a global manager 325 15.12 Upgrading NNMi in a global network management environment 327 15.13 Global network management and address translation protocol 328 16 NNMi IPv6 Management Feature 329

- 16.1 Overview of the NNMi IPv6 management feature 330
- 16.2 Prerequisites for using the NNMi IPv6 management feature 331
- 16.3 Licensing to use the NNMi IPv6 management feature 332
- Environment supported by the NNMi IPv6 management feature 16.4 333
- 16.4.1 Types of NNMi management servers and supported functions 333
- 16.4.2 Supported SNMP MIBs for IPv6 333
- 16.5 Installing NNMi and activating the IPv6 management feature 334
- 16.6 Deactivating the IPv6 management feature 335
- 16.6.1 IPv6 monitoring following deactivation of the IPv6 management feature 336
- IPv6 inventory following deactivation of the IPv6 management feature 16.6.2 336
- 16.6.3 Known issues when cleaning Up IPv6 inventory 336
- 16.7 Reactivating the IPv6 management feature 337

Part 5: High Availability Environment Configuration

17 NNMi Data Resilience 339

- 17.1 Approaches to NNMi data resilience 340
- 17.2 Comparison of approaches to NNMi data resilience 341

18 Configuring NNMi for Application Failover 342

- 18.1 Application failover overview 343
- 18.2 Application failover basic setup 344
- 18.2.1 Prerequisites for setting up application failover 344
- 18.2.2 Notes on application failover 345
- Configuring NNMi for application failover 18.3 347
- 18.3.1 Configuring application failover manually 347
- 18.3.2 Configuring application failover with the NNMi Cluster Setup Wizard 350
- 18.3.3 Setting application failover communications 351
- 18.4 Using the application failover feature 353
- 18.4.1 Application failover behavior 353
- 18.4.2 355 Application failover scenarios
- 18.4.3 ovstart and ovstop commands used on NNMi management servers configured for application failover 357
- 18.4.4 Application failover incidents 357
- 18.5 Returning to the original configuration following a failover 359

- 18.6 Disabling application failover 360
- 18.7 Administrative tasks and application failover 362
- 18.7.1 Upgrading NNMi (including applying a patch) 362
- 18.7.2 Starting, stopping, and restarting NNMi 362
- 18.7.3 Backing up and restoring NNMi 363
- 18.7.4 Modifying the NNMi settings 366
- 18.7.5 Changing the NNMi database password 368
- 18.8 Network latency/bandwidth considerations 369
- 18.8.1 Application failover and the NNMi database 369

19 Configuring NNMi in a High Availability Cluster 373

19.1	HA concepts 374
19.1.1	HA terms 375
19.1.2	NNMi HA cluster scenarios 375
19.1.3	Manpages 376
19.2	Verifying the prerequisites to configuring NNMi for HA 377
19.3	Notes about HA configurations 379
19.3.1	Notes about using related products 379
19.3.2	Notes about configuration tasks and operations 379
19.3.3	Other notes 380
19.4	Configuring HA 381
19.4.1	Configuring NNMi certificates for HA 381
19.4.2	Configuring NNMi for HA 381
19.4.3	Configuring NNMi for HA (Windows) 385
19.4.4	Configuring NNMi for HA (Linux) 392
19.5	Shared NNMi Data 401
19.5.1	Data on the NNMi shared disk 401
19.5.2	Replication of configuration files 402
19.6	Maintaining the HA Configuration 403
19.6.1	Placing NNMi in maintenance mode 403
19.6.2	Maintaining NNMi in an HA cluster 404
19.7	Unconfiguring NNMi from an HA cluster 408
19.7.1	Determining the active cluster node 408
19.7.2	Unconfiguring NNMi on the passive cluster node 408
19.7.3	Unconfiguring NNMi on the active cluster node 411
19.8	Troubleshooting the HA Configuration 415
19.8.1	Common configuration mistakes 415
19.8.2	HA resource testing 415
19.8.3	General HA troubleshooting 416
19.8.4	NNMi-specific HA troubleshooting 418
19.9	HA configuration reference 422
19.9.1	NNMi HA configuration files 422

19.9.2 NNMi-provided HA configuration scripts 422

19.9.3 NNMi HA configuration log files 423

Part 6: NNMi Maintenance

20	NNMi Backup and Restore Tools 425
20.1	Backup and restore commands 426
20.2	Backing up NNMi data 427
20.2.1	Backup type 427
20.2.2	Backup scope 427
20.3	Restoring NNMi data 430
20.3.1	Same-system restore 430
20.3.2	Different-system restore 431
20.4	Backup and restore strategies 432
20.4.1	Back up all data periodically 432
20.4.2	Back up data before changing the configuration 432
20.4.3	Back up data before upgrading NNMi or the operating system 433
20.4.4	Restore file system files only 433
20.5	Backing up and restoring the database 434
21	Maintaining NNMi 435
21.1	Administering access control lists for NNMi folders 436
21.2	Configuring node groups 437
21.3	Configuring node group map settings 438
21.4	Configuring communication settings 439
21.5	Administering a Custom Poller collection export 440
21.5.1	Changing the Custom Poller collections export directory 440
21.5.2	Changing the maximum amount of disk space for Custom Poller collections export 441
21.5.3	Changing the Custom Poller metric accumulation interval 441
21.6	Administering incident actions 442
21.6.1	Setting the number of simultaneous actions 442
21.6.2	Setting the number of threads for Jython actions 442
21.6.3	Setting the action server name parameter 443
21.6.4	Changing the action server queue size 443
21.6.5	Incident actions log 444
21.7	Overriding settings in the server.properties file 445
21.7.1	Override the browser locale setting 445
21.7.2	Configuring SNMP Set object access privileges 446
21.8	Administering SNMP traps 447
21.8.1	Configuring NNMi to authenticate SNMPv3 traps for nodes that are either managed by using SNMPv2 or SNMPv1 or that are not discovered 447
21.8.2	Block SNMPv1 or SNMPv2c Traps 448

21.8.3	Configuring timeframes within which the Causal Engine stops accepting traps 449
21.9	Blocking incidents using the trapFilter.conf file 451
21.10	Configuring character set encoding settings for NNMi 452
21.11	Modifying MIB Browser Parameters 453
21.12	Configuring NNMi to allow level 2 operators to delete nodes and incidents 454
21.13	Configuring NNMi to allow level 2 operators to edit maps 455
21.14	Configuring NNMi to allow level 1 operators to run status polls and configuration polls 456
21.15	Determining the original trap address from traps sent by a proxy SNMP gateway 458
21.15.1	Trap address ordering 459
21.16	NNMi NmsTrapReceiver process 460
21.16.1	Configuring the NmsTrapReceiver 460
21.16.2	Starting and stopping the NmsTrapReceiver process 460
21.17	Configuring HTTPS-only communication with the NNMi console 461
21.18	Configuring NNMi to require encryption for remote access 462
21.19	Configuring NNMi to preserve a previously supported varbind order 463
21.20	Configuring the auto-trim oldest SNMP trap incidents feature 465
21.20.1	Enabling the auto-trim oldest incidents feature (no incident archive) 465
21.20.2	Enabling the auto-trim oldest SNMP trap incidents feature (incident archive enabled) 466
21.20.3	Rotate archive files 467
21.20.4	Changing the maximum number of SNMP trap incidents to be saved 468
21.20.5	Monitoring the auto-trim oldest SNMP trap incidents feature 470
21.20.6	Disabling the auto-trim oldest SNMP trap incidents feature 470
21.21	Modifying NNMi normalization properties 472
21.21.1	Changing normalization properties following an initial discovery 473
21.22	Modifying the database port 474
21.23	NNMi self monitoring 475
21.24	Suppressing the use of discovery protocols for specific nodes 476
21.24.1	Suppressing the use of discovery protocol collections 476
21.25	Configuring actions for secondary root cause management events 478
21.26	Scheduling outages 479
21.27	Configuring sensor status 480
21.27.1	Configuring physical sensor status 480
21.27.2	Configuring node sensor status 482
22	Changing the NNMi Management Server 484
22.1	Best practices for preparing the NNMi configuration to be moved 485
22.2	Moving the NNMi configuration and database 486
22.3	Moving the NNMi configuration 487
22.4	Changing the IP address of a stand-alone NNMi management server 488
22.5	Changing the host name or domain name of an NNMi management server 489

23 NNMi Security 490

- 23.1 Providing a password for embedded database tools 491
- 23.2 Configure TLS Protocols 492
- 23.3 NNMi data encryption 493
- 23.3.1 Encryption and user account passwords 493

Part 7: Migration

24 Upgrading from NNMi Version 9, 10, or 11 495

- 24.1 Upgrading NNMi management servers 496
- 24.1.1 Upgrading NNMi management servers from version 12-10 496
- 24.1.2 Upgrading NNMi management servers from version 9, 10, or 11 496
- 24.2 Upgrading to a different NNMi management server 497
- 24.3 Upgrading global and regional managers from NNMi 12-10 498
- 24.3.1 NNMi versions supported by global network management 498
- 24.3.2 Global network management upgrade steps 498
- 24.4 Upgrading to NNMi 12-50 configured for application failover 499
- 24.4.1 Upgrading from NNMi 12-10 configured for application failover 499

25 Comparison with NNM Version 8 or Earlier 506

- 25.1 Network discovery 507
- 25.1.1 Key concepts of discovery 507
- 25.2 Status monitoring 509
- 25.2.1 Key concepts of status monitoring 509
- 25.3 Customizing event monitoring 511
- 25.3.1 Key concepts of event monitoring 511

26 Upgrading from NNM Version 8 or Earlier 513

- 26.1 Migration procedure 514
- 26.1.1 New NNM system 514
- 26.1.2 Migrating in phases 514
- 26.2 Phase 1: Migrating the SNMP information 516
- 26.2.1 Configuring SNMP access 516
- 26.2.2 Restricting name resolution 519
- 26.2.3 Customizing device profiles 520
- 26.3 Phase 2: Migrating discovery 522
- 26.3.1 Configuring a discovery schedule 522
- 26.3.2 Selecting a discovery method 524
- 26.3.3 Setting up auto-discovery rules 524
- 26.3.4 Adding seeded discovery 529
- 26.4 Phase 3: Migrating status monitoring 531
- 26.4.1 Specifying a polling interval 531

- 26.4.2 Selecting a polling protocol 532
- 26.4.3 Setting up important nodes 536
- 26.4.4 Excluding objects from state polling 538
- 26.5 Phase 4: Migrating event configuration and event reduction 539
- 26.5.1 Displaying traps from devices 539
- 26.5.2 Customizing the display of management events generated by NNMi 541
- 26.5.3 Setting up blocking, ignoring, or disabling of traps 542
- 26.5.4 Setting up automatic actions 542
- 26.5.5 Setting up additional (manual) actions 543
- 26.5.6 Event correlation: Event repetition 543
- 26.5.7 Event correlation: Rate calculation 544
- 26.5.8 Event correlation: Pairwise cancellation 545
- 26.5.9 Event correlation: ScheduledMaintenance 545

27 Migrating NNMi from an HP-UX or Solaris Operating System 546

- 27.1 Changing NNMi from HP-UX or Solaris to Linux 547
- 27.2 Changing NNMi from HP-UX or Solaris to Linux on Application Failover environment 549
- 27.3 Changing NNMi from HP-UX or Solaris to Linux on Application Failover environment on the global manager and regional managers 550

566

27.4 Changing NNMi from HP-UX or Solaris to Linux in a High Availability Cluster 551

Part 8: Integration with NNMi

NNMi Northbound Interface 28 552 28.1 Overview of the NNMi Northbound interface 553 28.2 Enabling the NNMi Northbound interface 554 28.3 Using the NNMi Northbound interface 555 28.3.1 Incident forwarding 555 28.3.2 Incident lifecycle state change notifications 556 28.3.3 Incident correlation notifications 556 28.3.4 Incident deletion notifications 557 28.3.5 Event forwarding filter 558 28.4 Changing the NNMi Northbound interface 559 Disabling the NNMi Northbound interface 560 28.5 28.6 Troubleshooting the NNMi Northbound interface 561 28.7 Application failover and the NNMi Northbound interface 562 28.7.1 Local Northbound application 562 Remote Northbound application 28.7.2 562 NNMi Northbound Interface Destination form reference 28.8 563 28.8.1 NNMi Northbound application connection parameters 563 28.8.2 NNMi Northbound interface integration content 564 28.8.3 NNMi Northbound interface destination status information

- 28.8.4 MIB information used by the NNMi Northbound interface 566
- 28.8.5 SNMP trap information used by the NNMi Northbound interface 566

Part 9: Integration

- 29 Integration with JP1/IM2 Intelligent Integrated Management Base 568
- 29.1 Integration with JP1/IM2 Intelligent Integrated Management Base 569

30 RESTful API 570

30.1 RESTful API 571

Appendixes 572

A	When NNMi Manpages Cannot Be Displayed (Linux) 573
В	List of MIBs Read During a New Installation 574
С	NNMi Environment Variables 581
C.1	Environment variables used in this manual 581
C.2	Other available environment variables 582
D	The Causal Engine and NNMi Incidents 584
D.1	Causal relationship analysis - advanced consideration 584
D.2	Causal Engine concept 584
D.3	Concept of status 585
D.4	About episodes 585
D.5	What does NNMi analyze? 586
D.6	Failure scenarios 588
D.7	Network configuration changes 609
D.8	NNMi management configuration changes 610
E	List of Ports Used by NNMi 612
F	Version Changes 617
F.1	Changes in version 12-50 617
F.2	Changes in version 12-10 617
F.3	Changes in version 12-00 618
F.4	Changes in version 11-50 620
F.5	Changes in version 11-10 621
F.6	Changes in version 11-00 624
F.7	Changes from version 10-10 to version 10-50 626
F.8	Changes from version 10-00 to 10-10 627
G	Reference Material for This Manual 631
G.1	Related publications 631
G.2	Conventions: Abbreviations for product names 631
G.3	Conventions: Acronyms 631
G.4	Conventions: KB, MB, GB, and TB 633
Н	Glossary 634

Index 642

Part 1: Preparation



Preinstallation Checklists

This chapter describes how to perform the preparations and checks required before you install NNMi.

Before installing NNMi, read the information about supported NNMi hardware and software in the NNMi Release Notes.

^{1.} Preinstallation Checklists

1.2 Preparing the preinstallation NNMi management server environment

An NNMi management server is a server on which the NNMi software is installed. Each NNMi management server must be a dedicated 64-bit machine. To learn more about hardware prerequisites, see 1.1 Checking the hardware and software.

Before you install NNMi on the NNMi management server, complete the checklist in Table 1-1.



Configure the remote desktop as described below before installing and configuring NNMi. This configuration increases the resources that Windows uses, so we recommend that you return to the original configuration after you complete these tasks, if necessary.

• Configuration path

Windows Server 2012 and later

Local Group Policy Editor[#] > Computer Configuration > Administrative Templates > Windows Components > Remote Desktop Services > Remote Desktop Session Host > Temporary folders #: To open Local Group Policy Editor, enter gpedit.msc in the Start window.

Configuration settings

Windows Server 2012 and later

Enable **Do not use temporary folders per session** and **Do not delete temp folder upon exit**. To apply these settings to the system, log off and then log on again.

Completed? (y/n)	NNMi management server preparation
	If you fail installing NNMi and install it again to the same environment, please refer to the manual deletion procedure in the appendix of the release note before installation.
	Make sure that the host name of the server where you plan to install NNMi is RFC-compliant.
	Host names are allowed to use alphanumerics (A to Z , a to z , 0 to 9), hyphens (-), and periods (.) (to demarcate domain names).
	Setup of host names that are not RFC-compliant (host names that use underscores (_), for example) might result in failure of the NNMi console connection or command execution.
	Make sure that the name of the local host can be resolved on the server on which NNMi is installed and that localhost is set up with the name resolved to 127.0.0.1.
	Windows
	Make sure that the C drive is used as the OS's system drive. NNMi cannot be installed in an environment where the system drive is not drive C .
	Windows
	If you have restrictive security settings in place, you might need to adjust the permission on the drive or drives on which you want to place the NNMi install and data directories. For details, see 1.5.1 Specifying disk drive security settings (Windows).
	Windows
	Check for the SNMP service; if installed, the SNMP trap service needs to be disabled on this server.
	Install and enable a supported Web browser. For details, see 1.1 Checking the hardware and software and 1.5.3 Enabling the Web browser for the NNMi console.

Completed? (y/n)	NNMi management server preparation
	Dynamic Host Configuration Protocol (DHCP) users: Makes sure that the NNMi management server is consistently assigned the same IP address.
	Disable anti-virus software until NNMi installation is complete. When NNMi installation is complete, restart the anti-virus software.
	 Linux Before you can install NNMi on a Linux server, the library files, command, and package listed below that are required by NNMi must be installed. Also, install the library files that have dependence relationships with these files: /lib64/libaio.so.1
	 /usr/lib64/libXtst.so.6 /usr/lib(4/libXi sp. 6
	• /usr/11064/110X1.so.6
	• net-tools package
	• upzip command
	• font config mackage
	 liberation-sans-fonts package[#] #: This package isn't necessary if the fc-list command shows one or more fonts.
	For RHEL 8.1 or later, CentOS 8.1 or later, or Oracle Linux 8.1 or later, additionally install the following. libnsl package
	For details, see 1.5.4 Installing required libraries in Linux.
	The database used by NNMi is PostgreSQL. If you install NNMi on a server on which PostgreSQL is already installed, make sure that there are no conflicts in the port settings. The port for PostgreSQL that NNMi uses is 5432/TCP. Therefore, change the port used by any existing PostgreSQL to a value other than 5432/TCP before you install NNMi. After installation, you can change the PostgreSQL port to be used by NNMi, if necessary.
	Do not install NNMi until you have verified that all ports used by NNMi are available. For a list of the ports used by NNMi and the direction in which data passes through a firewall, see E. List of Ports Used by NNMi.
	Do not block communication with the IP address of the local host, such as by using a firewall.
	Windows NNMi is installed in the language specified in the language settings in Format in Control Panel. NNMi cannot be used in any language other than the one specified in Format. If you want to change the language which is selected during OS installation, only changing from English to other languages is supported. Before you install NNMi, specify the following settings:
	Format, and make sure that the language set under Display language (in the Keyboards and Languages tab) matches the one set here.
	In an overwrite installation, the language setting specified here must match the setting for NNMi prior to the overwrite installation.
	2. In the Administrative tab, click Copy settings, and then click OK to copy the settings to the system account.
	3. In the Administrative tab, specify the language under Language for non-Unicode programs.
	Specify the language that was specified in Format in step 1.
	For the NNM1 management server's locale, specify one of the following:
	Ja_JP.UTF-8, C, OF Zh_CN.uti8
	Configuration information from earlier versions of NNM remains, even if NNM is removed. See the <i>Release Notes</i> for the earlier version of NNM to identify the older information, and then delete that information prior to installing NNMi.
	Windows Before installing NNMi, make sure that the Windows Services window (the window started from Control Panel > Administrative Tools > Services) is not running. If it is, close it.

Completed? (y/n)	NNMi management server preparation
	Windows If the sum of the character string set in the system environment variable Path and the lengths of the directory paths below is 950 bytes or more, the directory paths below might fail to be added to the system environment variable Path even if NNMi is successfully installed. %NnmInstallDir%bin\;
	%NnmDataDir%shared\nnm\actions\;
	If the directory paths above are not added to the system environment variable Path after installation is complete, add the directory paths manually to the system environment variable Path.
	Windows If you are installing NNMi using a path other than the default, you can use alphanumerics (A to Z, a to Z, 0 to 9), hyphens (-), periods (.), underscores (_), and single-byte spaces for the names of the install and data directories. Note that multiple consecutive single-byte spaces cannot be used. The maximum length of the absolute paths for these directories is 60 characters.
	Windows
	Do not specify paths that include junction points, such as <i>drive</i> : \Documents and Settings. Doing so might cause problems, such as temporary files not being deleted.
	Windows
	If you install NNMi in an environment in which environment variables %TEMP% and %TMP% have different values, installation might fail. Make sure that the values of %TEMP% and %TMP% are the same before installation. If they differ, set %TEMP% and %TMP% to the same values.
	Windows Do not set the following variables as environment variables:
	• LANG
	• Anything that begins with LC If another product sets these environment variables, it might not be compatible with NNMi. Installation might fail if NNMi is installed with these variables set.
	Windows
	If a Remote Desktop session host has been installed in Remote Desktop Services, the following setting is required before you install NNMi:
	• Executechangeuser/install to change to install mode.
	For details about this setting, see Help for the Remote Desktop session host.
	Windows If changes you have made to the system on which you plan to install NNMi require restarting the OS, do so prior to installing NNMi.
	For example, the OS must be restarted if the registry value below exists. If this value exists, NNMi might suspend the installation:
	HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Session Manager \PendingFileRenameOperations
	This registry value normally disappears when the OS is restarted.
	Windows NNMi uses a %TEMP% directory of up to 500 MB when it is installed or removed. The install or removal might fail without sufficient disk space.
	Linux NNMi uses a /tmp directory of up to 1 GB when it is installed or removed. The install or removal might fail without sufficient disk space.
	Windows During installation, under Local Group Policy Editor > Computer Configuration > Administrative Templates > Windows Components > Remote Desktop Services > Remote Desktop Session Host >

Completed? (y/n)	NNMi management server preparation
	Temporary folders , enable Do not use temporary folders per session and Do not delete temp folder upon exit . To apply these settings to the system, log off and then log on again.
	Windows
	Set Application Experience service to "Manual" if it is "Disabled". If Application Experience service is set as "Disabled", NNMi installation will fail.
	Linux If you are re-installing NNMi, also re-install all applications that require NNMi and configure them accordingly.
	Linux Install applications that require NNMi after the configuration of the environment that follows NNMi installation has been completed. Remember to execute the ovstop command before you install applications.
	Linux During NNMi installation, do not change the size of the terminal window in which Hitachi PP Installer is running. Doing so might prevent NNMi from installing properly.
	Linux
	NNMi requires a UDP reception buffer of 8 MB and a UDP transmission buffer of 2 MB.
	To change the settings for memory spaces allocated to buffers, edit the /etc/sysctl.conf file to add the following entries:
	<pre># NNM settings for UDP receive and send buffer sizes</pre>
	<pre>net.core.rmem_max = 8388608</pre>
	net.core.wmem_max = 2097152
	After editing the /etc/sysctl.conf file, apply the changes by restarting the OS or executing the /sbin/ sysctl -p command.
	Linux
	The value of kernel.shmmax or kernel.shmall might be too small. If they are, edit the /etc/ sysctl.conf file to add the following entries. We recommend a value of 64 GB.
	# NNM settings for embedded database
	kernel.shmmax = 68719476736
	kernel.shmall = 68719476736
	If you set the value of kernel.shmmax or kernel.shmall after editing the /etc/sysctl.conf file, apply the changes by restarting the OS or executing the /sbin/sysctl -p command.
	Linux
	The NNMi installation script automatically creates two groups (nmsgrp and nmsdb), two users (nmsproc and nmsdbmgr), and the corresponding \$HOME directories. This operation might fail for either of the following reasons:
	• Users and groups cannot be created because useradd or groupadd was disabled by the IT department.
	• The root user cannot create a \$HOME directory because the \$HOME directory exists on NFS.
	Installation stops whenever the NNMi installer is unable to create these groups, users, or directories. In such a case, you can create the users manually and start the installation.
	1. Create the nmsproc user in the nmsgrp group.
	Set the \$HOME directory to any directory that exists.
	2. Create the nmsdbmgr user in the nmsdb group.
	Set the \$HOME directory to any directory that exists.
	If you know that these operations will fail but you need to control user IDs, group IDs, or the locations of \$HOME, you can first create the groups, users, and \$HOME directories, and then start the installer. When the useradd command is used to create a user, the default home directory will be /home/user-name.
	Windows Server 2012 R2 Windows update program KB2919355 is required by NNMi must be installed.

NNMi uses Domain Name System (DNS) to determine relationships between host names and IP addresses. This can result in a large number of name service queries when auto-discovery is enabled.

Make sure that your DNS servers are well configured to prevent long delays when resolving name service queries. This means that the DNS server responding to NNMi name service queries has these characteristics:

- The DNS server is an authoritative server and does not forward DNS requests.
- The DNS server has consistent host name-to-IP address mappings and IP address-to-host name mappings.

If the network uses multiple DNS servers, all of them must respond consistently to all name service queries.

Important

Round-robin DNS (used to do load balancing of Web application servers) is not appropriate because any given host name can map to different IP addresses over time.

Note

To improve the response time for nslookup, deploy a secondary DNS service on the NNMi management server or another system on the same subnet as the NNMi management server. Configure this secondary DNS service to mirror the information from the primary DNS service. Another option is to use the following files instead of DNS in small environments:

• Windows

%SystemRoot%\system32\drivers\etc\hosts

• Linux

/etc/hosts

On the NNMi management server, make sure that the following are configured appropriately for your environment:

• The hosts file might take precedence with some OS configurations. Make sure that the hosts file contains a minimum of two entries:

127.0.0.1 localhost

NNMi-management-server-IP-address NNMi-management-server-name

NNMi-management-server-IP-address is the IP address of the fully qualified domain name (FQDN) of the NNMi management server. *NNMi-management-server-name* is the FQDN name for the NNMi management server set during installation.

• Windows

Make sure that all DNS servers used by the NNMi management server provide consistent host nameto-IP address mappings and IP address-to-host name mappings.

• Linux

Make sure that nslookup discovery conforms to the nslookup command discovery sequence set in the nsswitch.conf file.

Make sure that all DNS servers that you are aware of provide consistent host name-to-IP address mappings and IP address-to-host name mappings.

^{1.} Preinstallation Checklists

If you know that there are problems with the DNS configuration in your network domain (host names or addresses that do not resolve properly), instruct NNMi to avoid nslookup requests for unimportant devices. The benefits of doing this are as follows:

- Speed up spiral discovery
- Keep network traffic generated by NNMi to a minimum.

To identify problem devices to NNMi, create the two files listed below before configuring NNMi discovery. NNMi never issues a DNS request for host names or IP addresses identified in these files.

- hostnolookup.conf (enter fully qualified domain names or wildcards that identify groups of host names)
- ipnolookup.conf (enter IP addresses or wildcards that identify groups of IP addresses)

Use a text editor to populate the files. Place the files in the following locations on the NNMi management server:

Windows

%NnmDataDir%shared\nnm\conf\

%NnmDataDir% is the data directory specified during installation.

Linux

/var/opt/OV/shared/nnm/conf/

^{1.} Preinstallation Checklists

1.4 Preparing to use the NNMi Quick Start Configuration Wizard

You can run the Quick Start Configuration Wizard immediately after installation to configure NNMi in a limited (or test) environment. If you plan to use this wizard, complete the checklist in Table 1-2.

Table	1-2.	NNMi	Quick	Start	Configu	ration	Wizard	preinst	allation	checklist
Table	1-2.		QUICK	Start	Connigu	ration	VVIZaru	premsi	allation	CHECKIISL

Completed? (y/n)	Preparation for initial environment configuration
	Determine a limited IP address range for auto-discovery [#] . For details about the number of licenses (number of management nodes), see 2.3 Licensing NNMi.
	Determine IP addresses for discovery seeds. For details about seeds, see <i>About discovery seeds and auto-discovery rules</i> in 2.2 Using the Quick Start Configuration Wizard.
	Obtain the read-only SNMP community strings for the nodes within the discovery range from your network administrator.
	Determine a user name and password for an NNMi administrator account.

#

If use of network address translation (NAT) means that you will be managing areas in the network that contain duplicated IP addresses, select one address domain (an unduplicated address) that is to be detected by the Quick Start Wizard. Then, see *Overlapping Addresses in NAT Environments* in NNMi Help or 13. Managing Overlapping IP Addresses in a NAT Environment.

^{1.} Preinstallation Checklists

1.5 Additional information about installation

This section provides additional information about installing NNMi.

1.5.1 Specifying disk drive security settings (Windows)

To set disk drive security before installing NNMi, complete the following steps:

- 1. Open **Computer** to view your disk drives.
- 2. For the drive you plan to use for the NNMi installation, choose **Properties** in the right-click menu, and then click the **Security** tab.
- 3. Log on as a user with administrator privileges, making sure that **Full control** is allowed (either directly or derived through group membership).
- 4. On the Security tab, click Advanced to open Advanced Security Settings, and make sure the Applies To field for Administrators is set to This folder, subfolders, and files. If this is not the case, change the setting.
- 5. Make sure that **Full control** is allowed for the built-in **Local Service** user (either directly or derived through the local users group). If this is not the case, change the setting.
- 6. On the Security tab, click Advanced to open Advanced Security Settings, and make sure the Applies To field for the built-in Local Service user is set to This folder, subfolders, and files. If this is not the case, change the setting.
- 7. Apply your changes.
- 8. Proceed with the NNMi installation.

1.5.2 Obtaining or setting the official fully qualified domain name

NNMi users access NNMi by using the official fully qualified domain name (FQDN).

- 1. To determine the official FQDN of the NNMi management server, use one of the following methods:
 - Use the nnmofficialfqdn.ovpl command to display the value of the FQDN setting. For details, see the *nnmofficialfqdn.ovpl Reference Page*.
 - In the NNMi console, click **Help** > **System Information**. From the **Server** tab, find the value for the fully qualified domain name.
- 2. If you need to change the FQDN that was set during installation, use the nnmsetofficialfqdn.ovpl command.

For details, see the nnmofficialfqdn.ovpl Reference Page.

1.5.3 Enabling the Web browser for the NNMi console

Before you sign on to NNMi, make sure that the Web browser is configured to interact with the NNMi console. The following items must be enabled in the Web browser on each client machine that will access the NNMi management server:

- JavaScript
- · Pop-up windows from the NNMi management server

1. Preinstallation Checklists

JP1/Network Node Manager i Setup Guide

- Cookies from the NNMi management server
- ActiveX
- Automatic page loading
- If the IE ESC configuration is enabled in an environment that uses Internet Explorer, add about : blank to Trusted Sites.
- When Internet Explorer is used, disable the Compatibility View Setting for the NNMi management server.

Below are Web browser configuration examples.

Important

To complete the procedures below, you need to know the fully qualified domain name of the NNMi management server.

If your NNMi management server has more than one domain name, NNMi chooses one during the installation process. To determine which fully qualified domain name NNMi is using, run the nnmofficialfqdn.ovpl script. See the *nnmofficialfqdn.ovpl Reference Page* for more information.

(1) Mozilla Firefox

JavaScript is enabled by default in Mozilla Firefox. Privacy extensions are required to disable JavaScript. If an error is displayed in NNMi indicating that JavaScript is disabled, check the **Extensions** options in **Add-ons Manager** in Firefox to determine whether the privacy extensions are being used.

1. Enable pop-up windows by the following procedure:

a. In Mozilla Firefox, click Tools, and then choose Options.

- b. Click Content.
- c. Select the Block pop-up windows check box.
- d. Click Exceptions located next to the Block pop-up windows check box.

e. Add the fully qualified domain name of the NNMi management server to the list of allowed sites, and then click **Allow**.

f. Click Close.

- 2. Enable cookies by the following procedure:
 - a. In Mozilla Firefox, click Tools, and then choose Options.

b. Click Privacy.

- c. Move to History, and then select Remember history.
- 3. Restart the Web browser.

(2) Internet Explorer

To enable Internet Explorer as the Web browser for the NNMi console:

- 1. In Internet Explorer, click Tools, and then choose Internet Options.
- 2. On the Security tab, select the zone that includes the NNMi management server, and then click Custom Level.
- 3. In ActiveX controls and plug-ins, under Run ActiveX controls and plug-ins, select the Enable option.
- 4. Under Scripting, select the Enable option for Active scripting.

^{1.} Preinstallation Checklists
5. Under Others, select the Enable option for Automatic loading of pages.

6. On the Privacy tab Settings area, select one of the options between Accept All Cookies and Medium High.

Note

This setting affects the Internet zone only. If you are connecting to the NNMi management server over an intranet, this setting has no effect.

If, in the **Settings** area of the **Privacy** tab, the **Medium to High** option does not exist, you do not need to specify that option.

- 7. On the Privacy tab, select the Turn on Pop-up Blocker check box, and then click Settings.
- 8. Add the fully qualified domain name of the NNMi management server to the list of allowed sites.
- 9. Restart the Web browser.

For Internet Explorer 8, perform one of the following two procedures:

If the Internet Explorer Enhanced Security (IE ESC) configuration is enabled, perform the following steps in addition to the above:

- 1. In Internet Explorer, choose Tools, Internet Options, and then click the Security tab.
- 2. Add about:blank to the Trusted sites zone.

1.5.4 Installing required libraries in Linux

Before you can install NNMi on a Linux server, the library files, command, and package listed below that are required by NNMi must be installed. Also, install the library files that have dependence relationships with these files:

- /lib64/libaio.so.1
- /usr/lib64/libXtst.so.6
- /usr/lib64/libXi.so.6
- lsb_release command
- net-tools package
- unzip command
- fontconfig package
- liberation-sans-fonts package[#]
 #: This package isn't necessary if the fc-list command shows one or more fonts.

For RHEL 8.1 or later, CentOS 8.1 or later, or Oracle Linux 8.1 or later, additionally install the following.

• libnsl package

For details, see the NNMi Release Notes and the operating system documentation.

^{1.} Preinstallation Checklists

1.5.5 Setting the system account password

You can set or change the system account password by using the nnmchangesyspw.ovpl script. Follow these steps:

- 1. Use the ovstop -c command to stop the NNMi processes.
- 2. As an administrator, run the nnmchangesyspw.ovpl script to set the system password.
- 3. Use the ovstart -c command to start the NNMi processes.

For details, see the nnmchangesyspw.ovpl Reference Page.

^{1.} Preinstallation Checklists



Installing and Uninstalling NNMi

This chapter guides you through the process of installing and uninstalling NNMi. It also explains how to specify settings and how to obtain licenses after installation. Because the Release Notes also contain information about installation, please refer to the Release Notes as you read this chapter.

2.1 Installing NNMi

Windows

Before installing NNMi, complete the requirements listed in the preinstallation checklist, including disabling antivirus software. (See 1. Preinstallation Checklists.)

Linux

Before installing NNMi, complete the requirements listed in the preinstallation checklists. (See 1. Preinstallation Checklists.)

See the Release Notes for procedures for upgrading NNMi (including how to apply patches).

2.1.1 Installing NNMi (Windows)

To perform a new installation of NNMi in a Windows system, follow these steps:

Important

Configure the remote desktop as follows before you install and configure NNMi. This configuration increases the resources that Windows uses, so you can return to the original configuration after completing these tasks, if necessary.

• Configuration path

Windows Server 2012 and later

Local Group Policy Editor[#] > Computer Configuration > Administrative Templates > Windows Components > Remote Desktop Services > Remote Desktop Session Host > Temporary folders #: To open Local Group Policy Editor, enter gpedit.msc in the Start window.

• Configuration settings Windows Server 2012 and later

Enable **Do not use temporary folders per session** and **Do not delete temp folder upon exit**. To apply these settings to the system, log off and then log on again.

1. Log on as a user with administrator privileges to the system where you plan to install NNMi.

If UAC is enabled, a user who is not the built-in Administrator must be promoted to administrator.

- 2. Insert the NNMi installation media into the drive.
 - The Hitachi Program Product Installer window opens.
- 3. Start installation as indicated by the installer.
- 4. Enter information as indicated by the installer.

To use a default value, press the Enter key without entering any value. The default is the value shown in brackets.

• Specify the HTTP port number for the NNMi Web server.

Enter the HTTP port number for the NNMi Web server used to access NNMi. Enter a port number that is not being used by another program. The default value is 80.

An input example follows:

- ** Network Node Manager i Installer **
- * Starting NNMi installation.
- * Enter default port for HTTP server =>

```
* [80]
8004 •
```

• Specify the HTTPS port number for the NNMi Web server.

Enter the HTTPS port number for the NNMi Web server used to access NNMi. Enter a port number that is not being used by another program. The default value is 443.

An input example follows:

```
* Enter default port for HTTPS server =>
* [443]
* [443]
```

8443 🖊

• Specify the installation directory.

Enter the directory where you want to install the NNMi program. The default value is as follows: *drive*:\Program Files (x86)\Hitachi\Cm2NNMi\

An input example follows:

```
* Enter program install directory =>
```

```
* [C:\Program Files (x86)\Hitachi\Cm2NNMi\]
```

C:\Hitachi\Cm2NNMi\ 🚚

Important

NNMi includes 32 bit-compatible programs, so it cannot be installed in the *drive*:\Program Files\ folder of a 64-bit system.

We recommend that you install NNMi in *drive*:\Program Files (x86)\.

• Specify the data directory.

Enter the directory where you want to store NNMi configuration files and data such as databases and log files. The default value is as follows:

drive:\ProgramData\Hitachi\Cm2NNMi\

An input example follows:

```
* Enter program data directory =>
```

```
* [C:\ProgramData\Hitachi\Cm2NNMi\]
```

```
D:\NNMiData\ 🗸
```

• Check the display of entered data and make sure that installation has started.

The three data entries shown above are displayed. If there are no problems with the data displayed, enter yes to start installation. To change the entered data, enter no.

An input example follows:

- * http port : 80
- * https port : 443
- * install directory : C:\Hitachi\Cm2NNMi\
- * data directory : D:\NNMiData\
- * Do you start installation with above settings you entered ? (yes/no)

```
* If you need to change the settings, please enter no.
```

yes 🗸

• Confirmation of the completion of installation

If installation finishes normally, "Installation complete successfully." is displayed and the command prompt stops, so press the **Enter** key.

• Confirm whether to perform the preinstallation checks and continue the installation. The preinstallation checks will be performed first when installation starts.



In order to verify the items listed in Table 1-1: NNMi management server preinstallation checklist, the preinstallation check will verify that the ports used by NNMi are available.

Do not install NNMi until you have verified that all the ports used by NNMi are available. For a list of the ports used by NNMi and the direction in which data passes through a firewall, see E. List of Ports Used by NNMi.

If no problems are found in this preinstallation check, continue with the installation.

If there is a problem with the preinstallation check, the installer will output the problem and ask whether you want to continue the installation. Enter yes to continue the installation or no to terminate the installation.

Example of input when there is a problem in the preinstallation check:

```
* Starting NNMi Precheck ...
* TCP Port: [443] is used.
* UDP Port: [162] is used.
* NNMi Precheck result: NG
* There are some problem(s) with the settings.
* Do you want to continue NNMi installation ? (yes/no)
* If you enter no, the installation will stop.
```

no 🗸

Important

Do not use **Ctrl+Z** when entering this data. Entering **Ctrl+Z** suspends the installation. If the installation is suspended, resume the installation procedure from step 3.

Note that the installation takes ten to twenty minutes or more. Do not suspend the installation once it is underway. Interrupting the installation creates undesirable conditions that might render re-installation by normal means impossible.

Even if the precheck result is OK, this is not a guarantee that the installation will be successful. You must also make sure all the checklists in 1. Preinstallation Checklists are completed.

• Check the installation completion.

After installation complete rightly, a message that "Installation complete successfully." is output. Then press the **Enter** key.

• Check the installation results.

If the status shown in the last row of the log file %TEMP%\JP1NNMiInstaller.log is as follows, where a message stating that the process finished with [0] is output, installation was successful. If the status shows a value other than [0], installation might have failed.

```
[Trace] Process finished with [0]
```

Execute NNMi commands in a command prompt window that is opened after the installation has finished. Commands will not run correctly if they are executed from a window opened before the installation has finished, because settings required by NNMi commands, such as the NNMi environment variables, will not have been configured yet.

2.1.2 Installing NNMi (Linux)

To perform a new installation of NNMi in a Linux system:

- 1. Log in as a user with root privileges to the system where you plan to install NNMi.
- 2. Set a supported locale in the environment variable LC ALL, LANG:

```
# LC_ALL=ja_JP.UTF-8
# export LC_ALL
# LANG=ja_JP.UTF-8
# export LANG
```

For details about supported locales, see the Release Notes.

3. Place the NNMi media in the drive, and then mount the drive.

For details about how to mount drives, see the NNMi Release Notes and OS documentation.

4. Start Hitachi PP Installer.

Execute the following commands. mount-dir indicates the directory the drive is mounted as.

/mount-dir/X64LIN/setup /mount-dir

For details about how to start Hitachi PP Installer, see the NNMi Release Notes.

- 5. In the Hitachi PP Installer startup window, enter I to display a list of software that can be installed.
- 6. Move the cursor to JP1/Network Node Manager i, select it using the space bar, and then enter I. A message will appear asking you whether you want to continue the installation.
- 7. Enter y or Y.
- 8. Enter information as indicated by the installer.

To use default values, press the Enter key without entering any values. Defaults are the values shown in brackets.

• Specify the HTTP port number for the NNMi Web server.

Enter the HTTP port number for the NNMi Web server used to access NNMi. Enter a port number that is not being used by another program. The default value is 80.

An input example follows:

```
** Network Node Manager i Installer **
* Starting NNMi installation.
* Enter default port for HTTP server =>
* [80]
8004 .
```

• Specify the HTTPS port number for the NNMi Web server.

Enter the HTTPS port number for the NNMi Web server used to access NNMi. Enter a port number that is not being used by another program. The default value is 443.

An input example follows:

```
* Enter default port for HTTPS server =>
```

- * [443]
- 8443 🎝

```
2. Installing and Uninstalling NNMi
```

• Check the display of entered data and make sure that installation has started.

The data entered as shown above is displayed. If there are no problems with the data displayed, enter yes to start installation. To change the entered data, enter no.

An input example follows:

```
* http port : 80
* https port : 443
* Do you start installation with above settings you entered ? (yes/no)
* If you need to change the settings, please enter no.
```

yes 🗸

• Confirm whether to perform the preinstallation checks and continue the installation.

The preinstallation checks will be performed first when installation starts.

Note

In order to verify the items listed in Table 1-1: NNMi management server preinstallation checklist, the preinstallation check will verify that the ports used by NNMi are available.

Do not install NNMi until you have verified that all the ports used by NNMi are available. For a list of the ports used by NNMi and the direction in which data passes through a firewall, see E. List of Ports Used by NNMi.

If no problems are found in this preinstallation check, continue with the installation.

If there is a problem with the preinstallation check, the installer will output the problem and ask whether you want to continue the installation. Enter yes to continue the installation or no to terminate the installation.

Example of input when there is a problem in the preinstallation check:

```
* Starting NNMi Precheck ...
```

- * TCP Port: [443] is used.
- * UDP Port: [162] is used.
- * NNMi Precheck result: NG
- * There are some problem(s) with the settings.

```
* Do you want to continue NNMi installation ? (yes/no)
```

```
* If you enter no, the installation will stop.
```

no 🎝

Important

Note that the installation takes ten to twenty minutes or more. Do not interrupt the installation once it is underway. Interrupting the installation creates undesirable conditions that could render re-installation by normal means impossible.

Even if the precheck result is OK, this is not a guarantee that the installation will be successful. You must also make sure all the checklists in 1. Preinstallation Checklists are completed.

2.1.3 Operations after the installer finishes

This section describes tasks that must be performed after NNMi is installed. Perform these tasks no matter which OS you are using.

```
2. Installing and Uninstalling NNMi
```

(1) Exclude NNMi files and directories from the scan target

Anti-virus software might trigger the exclusive control of files and directories used by NNMi. This might cause a startup failure or abnormal termination of the NNMi service, or delay command execution. To perform a virus scan while NNMi is running, exclude files and directories within the following directories from the scan target:

Windows:

- NNMi installation directory
- NNMi data directory
- drive:\Program Files\Hitachi\Cm2NNMi (this directory might not exist.)
- *HA-mount-point*\NNM (In the case of cluster configuration)

Linux:

- /opt/OV
- /var/opt/OV
- HA-mount-point/NNM (In the case of cluster configuration)

If you perform a virus scan while NNMi is stopped, you need to restart NNMi after confirming that the virus scan for the above directories has been completed.

(2) Set the NNMi system account password

An account password is set for the initial sign-in to the NNMi console. Use the nnmchangesyspw.ovpl script to set the password. Run the nnmchangesyspw.ovpl script without arguments, and register a password as indicated by the displayed messages.

Note

The system account is a special administrator account created by the installation process. It is used when you sign in to the NNMi console for the first time. It is not normally used after the creation of a user with the administrator role on the NNMi console. The system account remains enabled after the installation finishes, but it is used only to execute on the command line or for restoration purposes. For details about how to set or change the system password, see 1.5.5 Setting the system account password.

(3) Set the language environment (Linux only)

Depending on the OS settings, rebooting the machine might automatically invoke the ovstart command with the setting LANG=C. In such a case, background processes will output messages in English. To prevent output of messages in English, configure the following settings to invoke the ovstart command in the supported locale at system startup.

• How to configure

- For distributions that uses systemd to manage

Make sure that of the following lines are entered before /opt/OV/bin/ovstart in the file /opt/OV/bin/ netmgt, and before /opt/OV/bin/ovstart nnmtrapreceivermd in the file /opt/OV/bin/ nettrap.

- Other distributions

Make sure that of the following lines are entered before /opt/OV/bin/ovstart in the file /etc/init.d/ netmgt, and before /opt/OV/bin/ovstart nnmtrapreceivermd in the file /etc/init.d/ nettrap.

• How to configure

Make sure that of the following lines are entered before /opt/OV/bin/ovstart in the file /etc/init.d/ netmgt, and before /opt/OV/bin/ovstart nnmtrapreceivermd in the file /etc/init.d/ nettrap. If necessary, enter one of the following lines:

```
LANG=ja_JP.UTF-8
export LANG
```

or

```
LANG=zh_CN.utf8
export LANG
```

(4) Check the maximum Java heap size

During installation, the maximum Java heap size (-Xmx) is set automatically according to the physical memory. Review the -Xmx value after consulting *Chapter 4. Memory and Disk Space Requirements* and *9.1 Systems* in the *Release Notes*. In addition, review the -Xmx value in the event of changes to the scale of the monitoring to be done with NNMi.

(5) Check the auto-trim configuration for incidents

Since NNMi 12-10, the auto-trim incidents is activated for new installations. Refer to 21.20 Configuring the auto-trim oldest SNMP trap incidents feature to review the presence of auto-trim configuration, the threshold for trimming, and the amount of deletions and settings to be deleted. If there is a change in the size of the object to be monitored, please review these configurations as well.

(6) Starting NNMi services

Execute the ovstart command to start NNMi services.

(7) Create an account to serve the administrator role

Sign in to the NNMi console and create an account to serve the administrator role.

1. A window for NNMi sign-in appears.

Enter the following URL in the window for entering the Web browser address.

http://fully-qualified-domain-name:port/nnm/

where *fully-qualified-domain-name* is the fully qualified domain name of the NNMi administrator server, and *port* is the HTTP port number of the NNMi Web server that was set during installation.

2. Enter the user name and password for the system account, and click the sign-in button.

- User name: system
- Password: The system account password created in (2) Set the NNMi system account password
- 3. Create the user accounts.

2. Installing and Uninstalling NNMi

```
JP1/Network Node Manager i Setup Guide
```

On the NNMi console, under **Configuration** workspace > **Security** > **User Accounts**, click the **New** icon. Enter a name and password and click the **Save and Close** icon to save the user account. For details, see *Configure User Accounts (User Account Form)* in NNMi Help.

Important

Between 1 and 40 characters can be entered for the name. Alphanumeric characters (A to Z, a to z, 0 to 9), periods (.), underscores (), at marks (@), and hyphens (-) can be used.

You can use one or more characters for the password. Supported characters include alphanumeric characters (A to z, a to z, 0 to 9) and single-byte symbols.

4. Map the administrator role to the user account.

On the NNMi console, under **Configuration** workspace > **Security** > **User Account Mappings**, click the **New** icon, and then specify the following parameters.

- User account: The user account created in step 3
- User group: NNMi administrator

Click the Save and Close icon to save the mapping. For details, see User Account Mapping Tasks in NNMi Help.

Important

Do not create a new user group. Instead, select from among the default user groups.

^{2.} Installing and Uninstalling NNMi

2.2 Using the Quick Start Configuration Wizard

This section guides you through some basic configuration tasks for NNMi. These tasks must be completed after you install NNMi. Very few parameters can be set from the Quick Start Configuration Wizard, which means that it is not possible to configure all settings necessary to start monitoring with NNMi. Hitachi normally recommends that you perform configuration from the NNMi console. You can use the Quick Start Configuration Wizard for initial setup (such as in a test environment), including:

- Configuring SNMP community strings
- Completing discovery of a limited range of network nodes

If use of network address translation (NAT) means that you will be managing areas in the network that contain duplicated IP addresses, select one address domain (an unduplicated address) that is to be detected by the Quick Start Wizard. Then see *Overlapping Addresses in NAT Environments* in the NNMi Help or 13. Managing Overlapping IP Addresses in a NAT Environment.

• Setting up an initial administrator account

Important

You cannot use the Quick Start Configuration Wizard to complete the SNMP Version 3 (SNMPv3) configuration. If you have devices that you prefer to monitor using SNMPv3, do the following:

- 1. Open the NNMi console.
- 2. In the Configuration workspace, select Communication Configuration.
- 3. Complete the SNMPv3 configuration.

After initial configuration, you can use the NNMi console for additional configuration tasks, such as adding nodes to the network topology and configuring monitoring. For details, see NNMi Help.

Note

Ë

About discovery seeds and auto-discovery rules

A discovery seed is a node that can help NNMi discover the network topology. For example, a seed might be a core router in your monitoring environment. Each seed is identified by an IP address or host name; see *Configure Auto-Discovery Rules* in NNMi Help.

- To configure discovery so that the devices that you specify as seeds become the starting point for additional discovery, create and configure auto-discovery rules; see *Specify Discovery Seeds* in NNMi Help.
- To configure discovery so that only the devices that you specify as seeds are discovered, do not create auto-discovery rules.

For overview information about the discovery process, see How Spiral Discovery Works in NNMi Help.

1. After the installation process finishes, launch the Quick Start Configuration Wizard as follows:

Run the Quick Start Configuration Wizard immediately after installation. To manually launch the Quick Start Configuration Wizard, go to the following URL:

http:// fully-qualified-domain-name:port/quickstart/

^{2.} Installing and Uninstalling NNMi

where *fully-qualified-domain-name* is the fully qualified domain name of the NNMi administrator server, and *port* is the port number that was set during installation.

If your NNMi management server has more than one domain name, NNMi chooses one during the installation process. To determine which fully qualified domain name NNMi is using, run the nnmofficialfqdn.ovpl script. For details, see the *nnmofficialfqdn.ovpl Reference Page*.

The NNMi Quick Start Configuration Wizard opens in a Web browser window.

- 2. Log on as follows:
 - User name: system
 - Password: Use the password for the system account that you created in (2) Set the NNMi system account password in 2.1.3 Operations after the installer finishes.
- 3. On the **Configure Community Strings** page, enter a community string for one of the nodes in the discovery range, and then click **Add**.

Quick Start Configuration Wizard		
Configure Community Strings NNM needs to collect some information from you before discovering and monitoring your network devices. We need to know the SNMP community strings for your network devices to start discovering and monitoring your network.	New Community String:	Add
	<previous next=""> Comm</previous>	Cancel
	Quick Start Configuration Wizard Configure Community Strings NNM needs to collect some information from you before discovering and monitoring your network devices. We need to know the SNMP community strings for your network devices to start discovering and monitoring your network.	Quick Start Configuration Wizard Configure Community Strings Information from you before discovering and monitoring your network devices to start oyour network devices to start network.

Note

NNMi automatically tries to match community strings to known devices. You do not need to manually associate each community string with a specific device.

4. Repeat step 3 until the **SNMP Community Strings** list includes the community strings for all nodes in the discovery range, and then click **Next**.

The SNMP community strings that you add here are saved in the NNMi database. In the NNMi console, the SNMP community strings are visible on the **Default SNMPv1/v2 Community Strings** tab of the **Communication Configuration** form.

5. On the **Configure Auto-Discovery Rules** page, associate the existing rule name with the **Included IP Address Range**. Enter the range of IP addresses for the discovery rule, and then click **Next**.

Examples of valid IP address ranges include:

- 10.1.1.*
- 10.1.1.1-99
- 10.10.50-55.*
- 10.1-7.1-9.1-9

2. Installing and Uninstalling NNMi

NNM Quick S	art Configuration Wizard
Add Community Strings Configure Auto Discovery Add Discovery Seeds Test Seeds Create Administrator Account Activate NNNi Community License Summary	If you configure a Rule Name and an Included IP Address Range, NMM will sutomatically discover the network devices contained in that IP address range. Enter an auto- discovery Rule Name and a range of addresse in the Included IP Address Range field. Rule Name: quickstart rule
	<previous next=""> Commit Cancel</previous>

6. On the **Configure Seeds** page, enter discovery seed information for your network, and then click **Add**. After that, click **Next**.

Enter discovery seeds in the form of IP addresses or fully qualified domain names. The network devices represented by these seeds help the NNMi spiral discovery process discover your network accurately.



You can use the nnmloadseeds.ovpl command to load seeds using a command line. For details, see the *nnmloadseeds.ovpl Reference Page*.

- 7. On the **Test Seeds** page, review the results of the communication tests. If any of the seed nodes cannot be reached with the community strings that you identified in step 3, click **Previous** to navigate to the **Configure Community Strings** page. Correct the community strings, and then click **Next**.
- 8. Repeat step 7 until all nodes can be reached, and then click Next.
- 9. On the **Configure Administrator Account** page, enter a **User Name**, and set the **Password** for a new account for administering the NNMi software. Then, click **Next**.
- 10. On the Summary page, review the information that you specified. Then, do one of the following actions:
 - To change any of the settings, click **Previous**.
 - To use the current settings, click Commit.

NNM Quick S	tart Configuration Wizard			
Add Community Strings Configure Auto Discovery Add Discovery Seeds Test Seeds Create Administrator Account Activate NNMi Community License Summary	Summary Review the displayed information and use the navigation buttons to make any corrections. Choose commit to apply and save these configuration changes.	Default Community Strings: Auto Discovery Rule: Included IP Range: Seeds: Administrative User Name: Activate Community License:	[nnmi] quickstart rule 102.168.100.1-255 [192.168.100.1] administrator No	
	[<previous next=""></previous>	Commit	Cancel

- 11. The **Wizard is complete** page indicates that you have successfully configured NNMi to discover a portion of your network. From here, do one of the following:
 - Click **Previous** to go back and make changes.
 - Click Launch UI to start the NNMi console user interface. To begin using NNMi, see 3. Getting Started with NNMi.



For Windows, after installation, restart any anti-virus software.

^{2.} Installing and Uninstalling NNMi

2.3 Licensing NNMi

If you do not have a permanent license key installed, the NNMi product includes a temporary license key that is valid for 60 days after you install NNMi. Therefore, obtain and install a permanent license key as soon as possible.

2.3.1 Preparing to install a permanent license key

The temporary license has a 250 node limit. If you have been running NNMi with a temporary license key, you might be managing more nodes than your permanent license supports. When the permanent license takes effect, NNMi automatically unmanages nodes of its choosing to achieve the license limit.

When tracking license information, note the following:

- Consumption: NNMi discovers and manages nodes up to the NNMi licensed capacity limit (rounded up):
 - VMware environments: Each device with a Device Profile of vmwareVM is equivalent to 1/10th node.
 - All other devices are equivalent to one discovered node.

If you want to control which nodes are no longer managed with the permanent license, use the NNMi console to delete less important nodes before installing your new license key.

(1) Checking the license type and the number of managed nodes

To determine the type of license that NNMi is using, follow these steps:

- 1. In the NNMi console, click Help > About Network Node Manager i.
- 2. In the About Network Node Manager i window, click Licensing Information.
- 3. Look for the value shown in the **Consumption** field.

This value is the number of nodes that NNMi is currently managing.

4. If your permanent license supports fewer nodes than NNMi is currently managing, use the NNMi console to delete less important nodes. For details, see *Delete Nodes* in NNMi Help.

2.3.2 Obtaining and installing a permanent license key

To request a permanent license key, gather the following information:

- The Software License Agreement
- The IP address of the NNMi management server
- Your company or organization information

For details about how to obtain and install a permanent license key, see the Release Notes.

2.3.3 Switching the temporary license

After NNMi is installed, a temporary license for NNMi is immediately applied.

To start the trial for functions of NNMi Advanced, it is necessary to switch the license key.

For details on how to switch between trial licenses for NNMi and for NNMi Advanced, see the Release Notes.

^{2.} Installing and Uninstalling NNMi

2.4.1 Removing NNMi (Windows)

1. Log on with administrator privileges to the system from which you plan to remove NNMi.

If UAC is enabled, a user who is not the built-in Administrator must be promoted to administrator.

- 2. Stop all NNMi services.
- 3. Under Control Panel > Programs and Features, select Network Node Manager and then click Uninstall or Change.
- 4. A dialog box is displayed asking whether you want to start the uninstallation. Enter yes to start uninstalling NNMi.

Example of input:

```
** Network Node Manager i Installer **
* Starting uninstallation ? (yes/no) =>
yes
```

5. After you have finished uninstalling NNMi, delete the NNMi installation directory and data directory.

Performing an uninstallation might not always delete the NNMi installation and data directories. In this case, delete them manually.

If default values were selected during installation, delete the following directories.

- *drive*:\Program Files (x86)\Hitachi\Cm2NNMi\
- *drive*:\ProgramData\Hitachi\Cm2NNMi\

6. Delete temporary directories and files.

Delete the following temporary directories and files that are created by NNMi.

The examples below show everything that might exist. There is no problem if some of the following do not exist. If you need the log output file for the uninstallation (NNMUninstall.log), make a copy of it before you delete the temporary file.

```
%TEMP%\HPOvInstaller\
%TEMP%\HPOvLic.log
%TEMP%\HPOvPerlA-install.log
%TEMP%\Install Autopass.log
%TEMP%\hsperfdata Administrator
%TEMP%\InstallerData
%TEMP%\JP1NNMiMIBLoad.log
%TEMP%\MicroFocusOvInstaller\
%TEMP%\NNMUninstall.log
%TEMP%\NNM X.X.X HPOvInstaller.txt (where X is one or more numeric characters)
%TEMP%\NNM X.X.X MicroFocusOvInstaller.txt (where X is one or more numeric charact
ers)
%TEMP%\nmscreatedb.log
%TEMP%\nnm hotfixes.log
%TEMP%\nnm installconfig vbs.log
%TEMP%\nnm premigration.log
%TEMP%\nnm_preinstallcheck_phaseI.log
%TEMP%\nnm preinstallcheck phaseII.log
%TEMP%\ovRemoveDir.exe
%TEMP%\ovDetach.exe
%TEMP%\ovinstallparams.ini
%TEMP%\ovCleanUp.bat
%TEMP%\persistent state
%TEMP%\preinstallcheck
```

```
%TEMP%\JP1NNMiInstaller.log
%TEMP%\JP1NNMiPostinstaller.log
%TEMP%\InstallScript.iap_xml
%TEMP%\nnm_preupgrade.log
%TEMP%\nnm_pre_dialogcheck.log
%TEMP%\OvLauncher.log
%TEMP%\OvLauncher.log
%TEMP%\nnm_pre-uninstall.log
%TEMP%\nnmilog\
drive:\ProgramData\apregid.com.hpe
```

7. Delete the environment variables.

Uninstalling NNMi does not delete the environment variables OVCSL_LOG, OVCSL_LOG_APPLICATION, and OVCSL_LOG_FILE, or *NnmInstallDir*bin\, which is added to the environment variable PATH when NNMi is installed. Delete them manually. Note that *NnmInstallDir* is the value set in environment variable NnmInstallDir.

2.4.2 Removing NNMi (Linux)

- 1. Log in as a user with root privileges to the system from which you plan to remove NNMi.
- 2. Stop all NNMi services.
- 3. Start the NNMi Uninstaller.

Execute the following commands to start Hitachi PP Installer.

```
# /etc/hitachi_x64setup
```

- 4. Following the instructions, select the NNMi Uninstaller, and perform the uninstallation.
- 5. After you have finished uninstalling NNMi, delete the NNMi installation directory and data directory.

Performing uninstallation might not always delete the NNMi installation and data directories. In this case, delete them manually.

Delete the following directories.

Installation directory

```
/opt/OV
```

Data directories

/var/opt/OV

6. Delete temporary directories and files.

Delete the following temporary directories and files that are created by NNMi.

The examples below show everything that might exist. There is no problem if some of the following do not exist. If you need the log output file for the uninstallation (NNMUninstall.log), make a copy of it before you delete the temporary file.

```
/var/tmp/HPOvPerlA-install.log
/var/tmp/jplnnmi
/var/tmp/JPlNNMiInstaller.log
/var/tmp/JPlNNMiPostinstaller.log
/var/tmp/rpm-tmp.xxx (where xxx is one or more alphanumeric characters)
/tmp/install.dir.xxx (where xxx is one or more numeric characters)
/tmp/ia_remove.shxxx.tmp (where xxx is one or more numeric characters)
/tmp/HPOvInstaller
/tmp/MicroFocusOvInstaller
/tmp/NNMUninstall.log
```

/tmp/NNM X.X.X HPOvInstaller.txt (where X is one or more numeric characters) /tmp/NNM X.X.X MicroFocusOvInstaller.txt (where X is one or more numeric character s) /tmp/debug /tmp/JP1NNMiMIBLoad.log /tmp/hsperfdata_bin /tmp/hsperfdata_nmsdbmgr /tmp/hsperfdata_nmsproc /tmp/hsperfdata root /tmp/nnm-premigration.log /tmp/nnm preinstallcheck phaseI.log /tmp/nnm_preinstallcheck_phaseII.log /tmp/ovinstallparams.ini /tmp/persistent_state /tmp/postInstall /tmp/postRemove /tmp/preInstall /tmp/preRemove /tmp/preinstallcheck /tmp/nnm-preupgrade.log /tmp/nnm_pre_dialogcheck.log /usr/share/apregid.com.hpe

2.5 Troubleshooting installation and initial startup

2.5.1 Installation problems

(1) Problem: NNMi installation requires more disk space than is currently available in the host system (Linux)

(a) Solution

When installing NNMi in Linux, you are not allowed to select the location where binary data is installed (\$OV_INST_DIR) or the location where data files are installed (\$OV_DATA_DIR). These locations are set as follows in the initial configuration:

- OV_INST_DIR=/opt/OV
- OV_DATA_DIR=/var/opt/OV

If there is insufficient disk space in either /opt/OV or /var/opt/OV, improve availability with either of the following methods:

- 1. If necessary, uninstall NNMi.
- 2. Create a symbolic link from the installation target to a partition that has sufficient disk space to install the binary data, and save the data files. The syntax for creating symbolic links is as follows:

```
ln -s large-disk /opt/OV
ln -s large-disk /var/opt/OV
```

Important

- Set the access permission for the parent directory at the installation site to 555 or higher.
- 3. Install NNMi.

(2) Problem: A message is displayed during installation indicating that the preinstallation procedure (phase II) has failed and the /tmp/ nnm_preinstall_phasell.log file needs to be checked for the details (Linux)

(a) Solution

The NNMi installation script automatically creates two groups (nmsgrp and nmsdb), two users (nmsproc and nmsdbmgr), and the corresponding \$HOME directories. This operation might fail for either of the following reasons:

- Users and groups cannot be created because useradd or groupadd was disabled by the IT department.
- The root user cannot create a \$HOME directory because the \$HOME directory exists on NFS.

Installation stops whenever the NNMi installer is unable to create these groups, users, or directories. In such a case, you can create the users manually and restart the installation.

^{2.} Installing and Uninstalling NNMi

- 1. Create the nmsproc user in the nmsgrp group. Set the \$HOME directory to any directory that exists.
- 2. Create the nmsdbmgr user in the nmsdb group. Set the \$HOME directory to any directory that exists.

If you know that these operations will fail but you need to control user IDs, group IDs, or the locations of \$HOME, you can first create the groups, users, and \$HOME directories, and then start the installer.

When the useradd command is used to create a user, the default home directory will be /home/user-name.

2.5.2 Initial startup problems

(1) Problem: The user cannot run the NNMi command line tools on Linux NNMi management servers

(a) Solution

Check whether system environment variable PATH includes /opt/OV/bin. If it does not, add /opt/OV/bin to system environment variable PATH.

(2) Problem: JBoss port contention

(a) Solution

By default, JBoss Application Server uses several ports for communication with NNMi. These ports are commonly used by other applications as well.

To resolve any port contention, follow these steps:

- 1. As a user with administrator privileges (Windows) or root privileges (Linux), open the following file in any text editor:
 - Windows

```
%NnmDataDir%Conf\nnm\props\nms-local.properties
```

%NnmDataDir% is the data directory specified during installation.

- Linux /var/opt/OV/conf/nnm/props/nms-local.properties
- 2. Modify the existing entries, replacing any conflicting port numbers with available port numbers.
- 3. Save the changes.
- 4. Execute the following commands to restart the NNMi services:

```
ovstop -c
ovstart -c
```

Note

For Windows, the ovstop and ovstart commands can be executed from the **Start** menu. For details about the ports used by NNMi, see the *nnm.ports Reference Page*.

(3) Problem: NNMi is not discovering nodes

(a) Solution

1. From the Workspace navigation panel, select the Configuration workspace.

- 2. From Configuration, open the Seeds view.
- 3. Check the values in the **Discovery Seed Results** column.

If the status of many of the discovered nodes is something other than **Node created**, then the NNMi discovery process was not successful.

If the status is **No SNMP response**, verify that you can ping the node, and that you can run the nnmsnmpwalk.ovpl command to obtain information from the node. For details, see the *nnmsnmpwalk.ovpl Reference Page*. If you cannot run these tools, check the following items:

- **a.** Ping the node to make sure it is responding.
- **b.** Make sure that the node has SNMP enabled.

c. Make sure that the node has your local management server on its access list of SNMP agents.

d. Be sure to configure the correct community strings for the nodes so that NNMi discovers them correctly. This information is listed on the **Communication Configuration** form on the **Default SNMPv1/v2 Community Strings** tab.

e. Make sure that there are no Access Control Lists configured on your routers, switches, or firewalls that might be limiting discovery.

For details, see Configure Discovery in NNMi Help.

(4) Problem: You cannot start the NNMi console when accessing a Windows NNMi management server

If you cannot start an NNMi console when pointing your browser to a Windows NNMi management server, a firewall might be blocking the HTTP port. To troubleshoot this problem, run the browser on the NNMi management server. If you can access the NNMi console from this browser, but remote browsers fail, you need to check your ports.

To remedy this problem, add the nmsas.server.port.web.http value shown in the %NnmDataDir%conf \nnm\props\nms-local.properties file to the list of allowed ports. For details, see the *nnm.ports Reference Page*.

%NnmDataDir% is the data directory specified during installation.

(5) Problem: The NNMi console does not open after a successful installation or upgrade of NNMi

Also, the following error message appears in latest copy of the incidentActions.*.*.log file (available in the /var/opt/OV/log/nnm/public directory on the NNMi management server):

SEVERE: com.hp.ov.nms.events.action.log.ActionLogger createActionServer:

Failed to get port number from /var/opt/OV/tmp/actionServer.port due to java.io.FileNotFoundException.

/var/opt/OV/tmp/actionServer.port.lock (Permission denied) :
java.io.FileNotFoundException:

/var/opt/OV/tmp/actionServer.port.lock (Permission denied)

(a) Solution

- 1. Run the following commands on the NNMi management server:
 - a./opt/OV/bin/ovstop
 - $b.\, \texttt{chown root:root /var/opt/OV/tmp}$
 - c.chmod 777 /var/opt/OV/tmp
 - d.chmod g+s /var/opt/OV/tmp
- 2. Look for the actionServer.port and actionServer.port.lock files in the /var/opt/OV/tmp directory. Delete those files if they exist.
- 3. Run the following command on the NNMi management server: /opt/OV/bin/ovstart

^{2.} Installing and Uninstalling NNMi

Part 2: Introduction



Getting Started with NNMi

This chapter provides information that you need to know before you begin using NNMi to manage your network. It includes a general overview of both how to access NNMi and how to specify network discovery settings. You can find more detailed information for operators and administrators in NNMi Help.

3.1 Accessing NNMi

Now that you have installed NNMi and completed post-installation configuration tasks, you can begin managing your network. All network monitoring and event-handling tasks can be accessed through the NNMi console, which opens in a Web browser.

To access the NNMi console, follow these steps:

- 1. Make sure that you are using a supported Web browser.
 - See 1.1 Checking the hardware and software.
- 2. Enable the Web browser for JavaScript, pop-up windows from the NNMi management server, and to accept cookies from the NNMi management server.

See 1.5.3 Enabling the Web browser for the NNMi console.

3. Enter the following URL into a Web browser window:

http://fully-qualified-domain-name:port/nnm/

where *fully-qualified-domain-name* represents the fully qualified domain name of the NNMi management server, and *port* is the port that JBoss Application Server uses for communicating with the NNMi console.

If your NNMi management server has more than one domain name, NNMi chooses one during the installation process. To determine which fully qualified domain name NNMi is using, run the nnmofficialfqdn.ovpl script. For details, see the *nnmofficialfqdn.ovpl Reference Page*.

If you cannot start an NNMi console when pointing your browser to an NNMi management server that is installed on a Windows operating system, you might have a Windows firewall on the NNMi management server that is blocking the http port. See (4) Problem: You cannot start the NNMi console when accessing a Windows NNMi management server.

4. In the NNMi sign-in window, enter your user account name and password, and then click **Sign In**. For details, see *About user accounts and roles* below.

About user accounts and roles

After installation, NNMi provides a special system account to be used to access NNMi for the first time. Do not use this system account for everyday use.

For everyday use, the NNMi administrator sets up an account for each user (or group of users) and assigns a preconfigured user role to each account. User roles determine who has access to the NNMi console, as well as which workspaces and actions are available to each user. NNMi provides the user roles listed below for NNMi console access. These roles are predefined by the program and cannot be modified:

- Administrator
- Operator level 2
- Operator level 1
- Guest

Before configuring NNMi sign-in access for your team, determine which pre-defined NNMi role is appropriate for each team member. The roles are hierarchical, meaning the higher level roles include all privileges of the lower-level roles in the hierarchy (Administrator is highest, Guest is lowest).

User accounts and roles, along with command-line access, are configured in the NNMi console. For details, see *Configuring Security* in NNMi Help.

NNMi provides an out-of-the-box https configuration using a self-signed certificate created during installation. For details about using a signed certificate from a Certificate Authority instead of the self-signed certificate, see 10. Working with Certificates for NNMi.

^{3.} Getting Started with NNMi

3.2 Accessing NNMi Help

NNMi Help describes how to use the NNMi console.

To access NNMi Help, click **Help** on the NNMi console menu bar, and then click one of the options above the first separator line on the menu.

Note

The NNMi console includes forms for entering information. The form name is in the upper left corner of the window. From any NNMi form, you can access the help information about that form. On the **Help** menu, click **Using the xyz form** where **xyz** is the title of the current form.

^{3.} Getting Started with NNMi

3.3 Configuring network discovery

As you begin to use NNMi to discover and manage your network, it is a good practice to start with a test network and configure NNMi to discover and manage a few nodes with only a few interfaces. The Quick Start Configuration Wizard (see 2.2 Using the Quick Start Configuration Wizard) provides an easy way to set up this type of small configuration. We recommend that you use the Quick Start Configuration Wizard immediately after installing NNMi.

As you become more familiar with NNMi, you will understand how its rich set of features applies to managing your network. You can expand the network topology that NNMi manages over time, systematically adding new discovery rules and putting new areas under management.

The topics in this section provide a brief overview of the configuration tasks that are required before initiating the discovery process. The checklist in the following table summarizes these tasks.

Completed? (y/n)	Task
	Verify that all nodes to be discovered are connected to the network and configured with a supported version of SNMP (SNMPv1, SNMPv2c, or SNMPv3).
	Obtain read-only community strings from your network administrator for the nodes that you plan to manage.
	Using the NNMi console, configure community strings as described in 3.3.1 Configuring community strings.
	Using the NNMi console, configure the spiral discovery process as described in 3.3.2 Configuring auto-discovery rules.
	Using the NNMi console, check the spiral discovery progress as described in 3.3.3 Checking discovery progress.

Table 3-1: Discovery configuration checklist

For details about the discovery process, see Discovering Your Network in NNMi Help.

3.3.1 Configuring community strings

To configure NNMi with community strings, follow these steps:

- 1. From the workspace navigation panel, select the **Configuration** workspace.
- 2. Open the **Communication Configuration** form, as shown here:

^{3.} Getting Started with NNMi

JP1/Network Node Manager i Setup Guide

Communication Config	uration * 🗙													
0 1 1 1														
Default SNMP Set	tings				Defa	ult SNMPv1A	2 Community	Strings	Default	SNMPv3 Se	ttings	Default D	evice Cred	. v
For more information, clic	ck here.			2	- Read	d Community	Strings	- (
Enable SNMP Address Rediscovery				1	For more in	formation, cli	ck here.							
Get-Bulk Enabled	✓				* 🖬	i I O I I	h 💼 👘				H 4	1 - 1 of	1 🔶	
* SNMP Timeout	5.00	Seconds •			▲ Read	Community	String							Orderin
Example: Timeout = 3 se communicate using SNM additional attempt, NNMi 9, and 12 for a total of 30	conds, Retries IP and waits 3 adds 3 secon I seconds. See	Count = 4. NN seconds for an ds before trying conline Help for	Mi attempts to answer. Each again, trying at 6, more information.		nnmi									
 SNMP Retries Count 	2													
 SNMP Port 	161													
SNMP Proxy Address														
SNMP Proxy Port														
SNMP Minimum Security Level	Community	•												
 Management Addr 	ress Selection													
If you choose Seed IP / M initial discovery. Otherwis Address, For more inform	Management II se, NNMi uses nation, click he	P, NNMi uses th the current Mar	e Seed IP only for nagement					То	tal: 1 🕴	Selected: 0	Filter	: OFF	Auto refr	esh: OFF
					• Write	e Community	String (Set Co	ommunity S	itring)					
* First Choice	Seed IP / Ma	anagement IP	-	F	for more in	formation, cli	ick here.							
* Second Choice	Interface Ma	tching 💌												
* Third Choice	Lowest Loop	back IP 👻			Write Con	nmunity	[
Interface Matching	100			~	Soring									

- 3. On the **Default SNMPv1/v2 Community Strings** tab, click the **New** icon.
- 4. On the **Default Read Community String** form, in the **Read Community String** box, enter a community string for one of the nodes in the discovery range, and then click the **Save and New** icon.
- 5. Repeat step 4 to enter all community strings for the nodes in the discovery range, and then click the **Save and Close** icon.
- 6. On the Communication Configuration form, click the Save and Close icon.

For details about setting up device community strings and loading community strings from a file, see *Configuring Communication Protocol* in NNMi Help.

3.3.2 Configuring auto-discovery rules

One of the most important network management tasks is keeping your view of the network topology up to date. NNMi maintains the topology through ongoing discovery of network nodes. The NNMi discovery process ensures that root cause analysis and the troubleshooting tools provide accurate information to resolve incidents. (See *Network discovery* in the *Note* below)

To configure auto-discovery rules, follow these steps:

- 1. From the workspace navigation panel, from the **Configuration** workspace, open **Discovery**.
- 2. Open the **Discovery Configuration** form.
- 3. Click the Auto-Discovery Rules tab, and then click the New icon.
- 4. On the Auto-Discovery Rule form, under Basics, enter the rule Name and Ordering information.

3. Getting Started with NNMi

The order is a numerical value that specifies the priority of this rule in comparison to other auto-discovery rules. For details, click **Help** > **Using the Auto-Discovery Rule form**.

5. Under Auto-Discovery Starting Point for this Rule, select the appropriate auto-discovery actions for this rule.

- 6. On the IP Ranges tab, click the New icon.
- 7. On the Auto Discovery IP Range form, enter the IP Range, and leave the Range Type set to Include in rule, and then click the Save and Close icon.
- 8. On the Auto-Discovery Rule form, click the Save and Close icon.
- 9. Repeat steps 3 through 8 until you have added all of the rules that you want to use.
- 10. On the **Discovery Configuration** form, click the **Save and Close** icon to save all new auto-discovery rules to the NNMi database.
- 11. From the Configuration workspace, open Discovery, and then click Seeds.
- 12. Click the New icon.
- 13. On the Discovery Seed form, enter a host name or IP address, and then click the Save and Close icon.
- 14. Repeat steps 12 and 13 until you have added all host names or IP addresses for discovery seeds.

To monitor the progress of discovery, see 3.3.3 Checking discovery progress.

For details about configuring discovery, see Configure Discovery in NNMi Help.

Note

Network discovery

NNMi collects information about the devices in your network (such as switches and routers) and proactively manages any devices that are important to you and your team. There are two discovery modes to choose from:

- Discovery seeds: You provide a list of devices and maintain total control over which devices NNMi discovers and monitors.
- Auto-discovery rules: You provide a list of addresses and host names as discovery seeds, and NNMi uses this information as starting points for extensive automatic discovery. You set limits on the NNMi discovery process by providing IPv4 address ranges and MIB II sysObjectIDs.

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

This dynamic discovery process continues over time. When things change in your network management domain, NNMi spiral discovery automatically updates information.

To learn more about network discovery, see Discovering Your Network in NNMi Help.

^{3.} Getting Started with NNMi

3.3.3 Checking discovery progress

After initiating the spiral discovery process, verify that the process is running correctly.

Note

Because spiral discovery is dynamic, NNMi discovers network nodes on a continuous basis. Whenever a new node is added to a discovery rule, NNMi discovers the node, collects topology information about the node, and begins to monitor the node.

There are several ways to gauge discovery progress. Perform any of the following actions to check the discovery progress:

- During discovery, check the status of seeds by using **Configuration** > **Discovery** > **Seeds**. Review the status information in the **Discovery Seed Results** column. When discovery is nearing completion, the majority of the nodes have the **Node created** status.
- During discovery, check the discovery progress by using Help > System Information, and then click the Database tab. Check the Database Object Counts several times during a one-hour period. The numbers in the Nodes, SNMP Agents, Interfaces, IP Addresses, and L2 Connections fields will stabilize. If these numbers are no longer increasing in value over the sampling period, then discovery is complete.
- During discovery, from the NNMi console, in the **Inventory** workspace, select **Nodes**. Check the value in the **Total** field several times during a one-hour period. If this number is no longer increasing in value over the sampling period, then discovery is complete.
- During discovery, from the NNMi console, check the discovery progress by clicking Tools > NNMi Self-Monitoring Graphs > Discovery Progress.
- During discovery, from the NNMi console, check the discovery progress by clicking Tools > Status Distribution Graphs > Node Status.
- During discovery, from the NNMi console, in the **Topology Maps** workspace, click **Network Summary**. Watch the map grow in complexity during a one-hour period. If the map growth slows and then stops over the sampling period, discovery is complete.



If you suspect a problem with discovery, see (3) Problem: NNMi is not discovering nodes.

^{3.} Getting Started with NNMi

Part 3: Configuration



General Concepts for Configuration

This chapter provides an introduction to concepts for configuration. More details are provided later in this manual beginning in Chapter 5. This chapter also contains some best practices that apply to all NNMi configuration areas.

4.1 Task flow model



The chapters in Part 2, Configuration of this manual provide information that is relevant to the following task flow:

- 1. **Concepts** -- Gain a general understanding of the configuration area. The information in this guide supplements the information in NNMi Help.
- 2. **Plan** -- Decide how you want to approach the configuration. This is a good time to begin or to 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. For specific procedures, see NNMi Help.
- 4. Evaluate -- Examine the results of your configuration on the NNMi console. Adjust the configuration as necessary to achieve the desired results.
- 5. Tune -- Optional. Adjust the configuration to improve NNMi performance.

^{4.} General Concepts for Configuration

4.2 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 details about how to use these commands, see the appropriate reference pages.

The nnmconfigexport.ovpl command does not retain SNMPv3 credentials. For details, see the *nnmconfigexport.ovpl Reference Page*.

^{4.} General Concepts for Configuration

4.3 Best practice: Use the Author attribute

Many NNMi configuration forms include the Author attribute.

As you use these forms to create or modify configurations, set in the Author attribute 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.

When you upgrade NNMi, the installer does not overwrite any configurations whose author value has been set by the user.

^{4.} General Concepts for Configuration

Some NNMi console forms use a transactional 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 on the NNMi console. If you close a form that contains unsaved changes, NNMi warns you about the unsaved changes and gives you a chance to cancel the close.

^{4.} General Concepts for Configuration
4.5 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 at other levels 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.

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

- For ordering menus and menu items, in which case you specify an order for applicable menus that is appropriate within the local context.
- For ordering topology maps on the Node Group Map Settings form, which sets the order of items in the Topology Maps workspace.

For specific information about how the Ordering attribute affects a given configuration area, see NNMi Help for that area.



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

^{4.} General Concepts for Configuration

4.6 Node groups and interface groups

For node and interface groups, you can set a filter to narrow the information to be displayed in a view. For example, if the node group **Important Cisco router** is set and used as a filter, you can display only the applicable routers in the view.

Node groups can be used for any or all of the following purposes:

- Monitoring settings
- Incident payload filtering
- Table filtering
- Customizing map views
- Filtering the nodes passed from a regional manager to the global manager for the global network management feature

Interface groups can be used for any or all of the following purposes:

- Excluding interfaces from discovery
- Monitoring settings
- Incident payload filtering
- Table filtering

You can create a hierarchy of node groups based on any filterable attributes to control map view drill-down, monitoring, or both settings' inheritance.

4.6.1 Group overlap

Regardless of the intended uses for 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 all of them OUT OF SERVICE together for an IOS upgrade. You can do this with a vendor filter.

^{4.} General Concepts for Configuration

• 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.

4.6.2 Node group membership

NNMi determines node group membership by comparing each discovered node to each of the configured node groups.

• All nodes specified on the Additional Nodes tab are members of the node group.

Note

Avoid using the **Additional Nodes** tab to add nodes to a node group, because doing so consumes excessive resources on the NNMi management server.

- All nodes that are members of at least one node group specified on the **Child Node Groups** tab are members of the node group.
- Any node that matches one or more entries (if any exist) on the **Device Filters** tab and the filter specified on the **Additional Filters** tab is a member of the node group.

(1) 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.

^{4.} General Concepts for Configuration

JP1/Network Node Manager i Setup Guide

Note

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.

An attempt to save a child node group with a circular reference node group set will fail with a warning.

(2) Device filters

During discovery, NNMi collects direct information through SNMP queries and derives information other than that through device profiles. For details, see 6.1.1 Device profiles and device attributes. 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
- Device profile

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. You can also group all devices of a type, such as routers, across vendors.

(3) Additional filters

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

- hostname (host name)
- 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)
- isSnmpNode (agent enabled)
- isNnmSystemLocal (NNMi management server)
- sysOidNode (system object ID)
- devCategoryNode (device category)
- devVendorNode (device vendor)
- devFamilyNode (device family)
- nnmSystemName (host name, case sensitive)

^{4.} General Concepts for Configuration

- nodeName (node name)
- securityGroupName (security group name)
- securityGroupUuid (security group UUID)
- tenantName (tenant name)
- tenantUuid (tenant UUID)

Filters can include the AND, OR, NOT, EXISTS, NOT EXISTS, and grouping (parentheses) operations. For details, see *Specify Node Group Additional Filters* in NNMi Help.

You can check capabilities by examining the node details from a device that has already been discovered.

(4) Additional nodes

It is better to use **Additional Filters** to qualify nodes for node groups. If the network contains critical devices that are too difficult to qualify using filters, add them to a group by individual host name. Only add nodes to a node group by individual host names as a last resort.

Note

Avoid using the **Additional Nodes** tab to add nodes to a node group, because doing so consumes excessive resources on the NNMi management server.

4.6.3 Node group status

NNMi determines the status of a node group using one of the following algorithms:

- Set the node group status to match the most severe status of any node in the node group. To use this approach, select the **Propagate Most Severe Status** check box on the Status Configuration form.
- Set the node group status using the thresholds set for each target status. For example, the default threshold for the target status of Minor is 20%. NNMi sets the status of the node group to Minor when 20% (or more) of the nodes in the node group have Minor status. To use this approach, clear the **Propagate Most Severe Status** check box on the **Status Configuration** form. You can change the percentage thresholds for the target thresholds on the **Node Group Status Settings** tab of this form.

Because status calculations for large node groups can be resource-intensive, the default for calculation of node group status is set to off for new installations of NNMi. You can enable status calculation with the **Calculate Status** check box on the **Node Group** form for each node group.

4.6.4 Interface groups

Interface groups filter interfaces within nodes by ifType or by other attributes, such as ifAlias, ifDesc, 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.

^{4.} General Concepts for Configuration

Important

Under the following conditions, interfaces in an interface group are not always excluded initially during discovery:

- The interface group was created by filtering on one or more interface capabilities in the interface group definition.
- The interface group is specified in the **Excluded Interfaces** discovery configuration option. After the interface capabilities are applied to an interface in the interface group, that interface will be excluded when the exclusion filter is re-applied during a rediscovery.

For details about the interface capabilities provided by NNMi and the **Excluded Interfaces** discovery configuration option, see the NNMi online Help for Administer.

^{4.} General Concepts for Configuration

4.7 Node/interface/address hierarchy

NNMi assigns monitoring settings in the following manner:

1. Interface settings -- NNMi monitors each 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.

^{4.} General Concepts for Configuration

4.8 Resetting the NNMi configuration and database

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 details about the commands identified in this procedure, see the appropriate reference pages.

To redo the configuration:

1. (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.

The nnmconfigexport.ovpl command does not retain SNMPv3 credentials. For details, see the *nnmconfigexport.ovpl Reference Page*.

2. (Optional) Use the nnmtrimincidents.ovpl command to archive the NNMi incidents.

Specify the -archiveOnly option because the default is that the nnmtrimincidents.ovpl command does not archive incidents. For details, see the *nnmtrimincidents.ovpl Reference Page*.

3. Stop the NNMi services with the following command:

ovstop -c

4. (Optional) Back up the existing database.

Because this procedure deletes the database, before proceeding you might want to use the following command to back up the existing database:

nnmbackup.ovpl -type offline -target backup-directory

5. Drop and re-create the NNMi database.

nnmresetembdb.ovpl -nostart

6. Start the NNMi services with the following command:

ovstart -c

NNMi now has only the default configurations as if you had just installed the product on a new system.

- 7. Start configuring NNMi. Do one of the following:
 - Use the Quick Start Configuration Wizard.
 - Enter information into the Configuration workspace on the NNMi console.
 - Use the nnmconfigimport.ovpl command to import some or all of the NNMi configuration that you saved in step 1.

Important

If you use the nnmconfigimport.ovpl command to import a large number of settings (such as settings for 9,500 node groups and 10,000 incidents), consider using the -timeout option to adjust the import transaction's timeout value to be greater than the default value of 60 minutes (3,600 seconds). For details, see the *nnmconfigimport.ovpl Reference Page*.

^{4.} General Concepts for Configuration



NNMi Communications

NNMi uses both Simple Network Management Protocol (SNMP) and Internet Control Message Protocol (ICMP ping) 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 discovery and state polling. NNMi applies the appropriate values for each device when making queries for discovery or polling. Thus, if you configure NNMi to disallow SNMP communication within some region of your network, neither NNMi discovery nor NNMi state polling can send SNMP requests to that region.

5.1 Concepts for communications

NNMi uses SNMP and ICMP primarily in a request-response manner. Responses to ICMP ping requests verify address responsiveness. Responses to SNMP requests for specific MIB objects provide more comprehensive information about a node.

Note

If Web Agents are configured (in addition to SNMP Agents), NNMi can use additional protocols. For example, SOAP protocol for VMware environments.

The following concepts apply to NNMi communications configuration:

- Levels of communication configuration
- Network latency and timeouts
- SNMP access control
- SNMP version preferences
- Management address preferences
- Polling protocols
- nnmsnmp*.ovpl commands

5.1.1 Levels of communication configuration

NNMi communication configuration provides the following levels:

- Specific nodes
- Regions
- Global defaults

At each level you can configure access credentials, timeout and retry values, management protocol enablement (for example, for ICMP and SNMP), and management protocol access settings (for example, SNMP). If you leave settings blank at one level, NNMi applies the next level of defaults.

When communicating with a given node, NNMi applies the configuration settings as follows:

- 1. If the node matches a specific node configuration, NNMi uses any communication values in that configuration.
- 2. If there is no such node as in 1 above, NNMi determines whether the node belongs to any regions. Because regions might overlap, NNMi uses the matching region with the lowest ordering number. NNMi uses the values specified for that region to fill in the blanks left from the applicable specific node setting (if any). The settings for additional regions with a higher ordering number are not considered.
- 3. If there is no such settings as in 1 and 2, NNMi uses the global default settings to fill in the remaining blanks.

The values used for management protocol communication with a particular device might be built up cumulatively until all required settings are determined.

5.1.2 Network latency and timeouts

Normal network latency influences the amount of time the NNMi management server must wait to get answers to ICMP and SNMP queries. Different areas of a network customarily have different turnaround times. For example, the local network where the NNMi management server resides could provide nearly instantaneous response, while responses from a device in a remote geographical region accessed through a dial-up wide area link would typically take much longer.

In addition, heavily-loaded devices might be too busy to respond to ICMP or SNMP queries immediately. When deciding which timeout and retry settings to configure, consider these latency concerns.

You can configure specific timeout and retry settings for both network regions and specific devices. The settings you choose determine how long NNMi waits for an answer and how many times NNMi requests data before abandoning the request when no answer is received.

For each request retry, NNMi adds the configured timeout value to the previous timeout value. Thus, the pause gets longer after each retry. For example, when NNMi is configured to use a timeout of five seconds and three retries, NNMi waits five seconds for a response to the first request. If there is no response, NNMi waits for 10 seconds for a response to the first request retry, and 20 seconds for a response to the third request retry before giving up until the next polling cycle.

5.1.3 SNMP access control

Communication with SNMP agents on managed devices requires access control credentials:

• SNMPv1 and SNMPv2c

A community string in each NNMi request must match a community string configured in the responding SNMP agent. All communication passes through the network in clear text (no encryption).

• SNMPv3

Communication with the SNMP agent complies with the user-based security model (USM). Each SNMP agent has a list of configured user names and their associated authentication requirements (the authentication profile). Formatting of all communication is controlled through configuration settings. NNMi SNMP requests must specify a valid user and follow the authentication and privacy controls configured for that user.

- Authentication protocol uses no message authentication or uses hash-based message authentication code (HMAC) with your choice of either HMAC-MD5-96 or HMAC-SHA-1.
- Privacy protocol uses no encryption or uses a symmetric encryption protocol with your choice of DES-CBC, TripleDES, AES-128, AES-192, or AES-256.

Because DES-CBC is regarded as a weak cipher, we recommend if you are using DES-CBC that you choose a stronger cipher. If you will be configuring SNMPv3 communication on a node managed by NNMi, we recommend that you not use DES-CBC.

To change the encryption method:

- 1. In the NNMi console, click the Configuration workspace.
- 2. Expand the **Incidents** folder.
- 3. Expand the **Trap Server** folder.
- 4. Click Trap Forwarding Configuration.
- 5. From the Privacy Protocol list, select a stronger encryption method.

NNMi supports the specification of multiple SNMP access control credentials for a region of your network (defined through IP address filters or host name filters). NNMi attempts communication with a device in that region by trying in parallel all configured values at a given SNMP security level. You can specify the minimum SNMP security level that NNMi is to use in that region. NNMi uses the first value returned by each node (response from the device's SNMP agent) for discovery and monitoring purposes.

In the default HA environment, the SNMP source address is set to the physical cluster node address. To set the SNMP source address to NNM_INTERFACE (the virtual IP address), edit the ov.conf file to set the IGNORE NNM IF FOR SNMP value to OFF (by default, it is ON).

5.1.4 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 network environment.

The first SNMP response NNMi receives for a particular node determines the communication credentials and SNMP version used by NNMi for communication with that node.

The SNMP version selection for a node plays a role in NNMi accepting traps from that node:

- If the source node or source object of the incoming trap has been discovered by NNMi using SNMPv3, NNMi accepts incoming SNMPv1, SNMPv2c, and SNMPv3 traps.
- If the source node or source object of the incoming trap has been discovered by NNMi using SNMPv1 or multiple SNMPv2cs, NNMi discards incoming SNMPv3 traps.

If these traps must be received, follow the procedure described in 21.8.1 Configuring NNMi to authenticate SNMPv3 traps for nodes that are either managed by using SNMPv2 or SNMPv1 or that are not discovered.

You specify the minimum level of SNMP version and security settings that are acceptable in each area of your network. The options for the **SNMP Minimum Security Level** field are as follows:

• Community Only (SNMPv1 only)

NNMi attempts to communicate using SNMPv1 with the configured values for community strings, timeouts, and retries. NNMi does not try any SNMPv2c or any SNMPv3 settings.

• Community Only (SNMPv1 or v2c)

NNMi attempts to communicate using SNMPv2c with the configured values for community strings, timeouts, and retries. If there is no response to any community string using SNMPv2c, NNMi attempts to communicate using SNMPv1 with the configured values for community strings, timeouts, and retries. NNMi does not try any SNMPv3 settings.

• Community

NNMi attempts to communicate using SNMPv2c with the configured values for community strings, timeouts, and retries. If there is no response to any community string using SNMPv2c, NNMi attempts to communicate using SNMPv1 with the configured values for community strings, timeouts, and retries. If none work, NNMi tries SNMPv3.

• No Authentication, No Privacy

For users with no authentication and no privacy, NNMi attempts to communicate using SNMPv3 with the configured values for timeouts and retries. If none work, NNMi tries users with authentication and no privacy followed by users with authentication and privacy, if necessary.

• Authentication, No Privacy

^{5.} NNMi Communications

For users with authentication and no privacy, NNMi attempts to communicate using SNMPv3 with the configured values for timeouts and retries. If none work, NNMi tries users with authentication and privacy.

• Authentication, Privacy

For users with authentication and privacy, NNMi attempts to communicate using SNMPv3 with the configured values for timeouts and retries.

5.1.5 Management address preferences

A node's *management address* is the address NNMi uses to communicate with the node's SNMP agent. You can specify the management address for a node (in the specific node settings), or you can let NNMi choose an address from the IP addresses associated with the node. You can fine-tune this behavior in the discovery configuration settings by excluding certain addresses from discovery. For details about how NNMi determines the management address, see *Node Form* in NNMi Help.



To discover hypervisors NNMi requires the node name rather than the management address.

NNMi discovers and monitors devices on an ongoing basis. After the first NNMi discovery cycle, the **Enable SNMP** Address Rediscovery field controls NNMi behavior when previously discovered SNMP agents quit responding (for example, when you reconfigure the device's SNMP agent).

- If the **Enable SNMP Address Rediscovery** check box is selected, NNMi retries any configured values in search of one that works.
- If the **Enable SNMP Address Rediscovery** check box is cleared, NNMi reports the device as **Down** and does not attempt to find another communication configuration setting for that device.

The Enable SNMP Address Rediscovery check box is available at all levels of communication configuration.

The **Discover Any SNMP Device** and **Discover Non-SNMP Devices** auto-discovery rule configuration fields influence the way NNMi uses SNMP. For details, see *Configure Basic Settings for the Auto-Discovery Rule* in NNMi Help.

5.1.6 SNMPv3 traps and informs

When NNMi uses SNMPv3 to communicate with a device, it uses a discovery process to identify the device's engine ID, boot count, and engine time. NNMi then uses this information, together with the configured user and protocol details, to start sending messages to the device.

A device might not have the NNMi information when it needs to send a trap to NNMi, and because a trap is a singlepacket transaction, it has no way to obtain the necessary information. In such a case, the device uses its own engine ID, boot count, and engine time in the trap, together with the user name and protocol details. These device details must be the same as those configured for the device in NNMi. You cannot configure multiple SNMPv3 users for a single device in NNMi.

An inform is an acknowledged packet, so this is more like an SNMP request that NNMi would make to the device except, this time, it is the device initiating the first packet and NNMi responding with an acknowledgment. The device, therefore, performs the discovery to NNMi to learn NNMi's engine ID, boot count, and engine time. The user name and protocol configuration that the device uses must match what is configured in the NNMi trap forwarding configuration; this is, in effect, NNMi's SNMPv3 agent configuration.

^{5.} NNMi Communications

JP1/Network Node Manager i Setup Guide

5.1.7 Polling protocols

You can prevent NNMi from using SNMP or ICMP in portions of your network (for example, when firewalls in your infrastructure prohibit ICMP or SNMP traffic).

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

- The optional auto-discovery rule Ping sweep feature cannot locate additional nodes in that region of your network. All nodes must either be seeded or available through answers to MIB object requests, such as neighbor's ARP cache, Cisco Discovery Protocol (CDP), or Extreme Discovery Protocol (EDP). Wide area network devices might be missed unless you seed every one of them.
- The State Poller cannot monitor devices that are not configured to respond to SNMP requests.
- Operators cannot use Actions > Node Access > Ping to check device reachability during troubleshooting.

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

- Discovery cannot gather any information about the devices except that they exist. All devices receive the **No SNMP** device profile.
- Discovery cannot find additional neighboring devices through queries. All devices must be directly seeded.
- Discovery cannot gather connectivity information from the devices, so they appear unconnected on NNMi maps.
- For devices with the **No SNMP** device profile, the State Poller respects the defaults of monitoring that device using only ICMP (ping).
- The State Poller cannot gather component health or performance data from the devices.
- The Causal Engine cannot contact the devices to perform neighbor analysis and locate the root cause of incidents.

5.1.8 nnmsnmp*.ovpl commands

The nnmsnmp*.ovpl commands look up the values for unspecified device communication settings in the NNMi database. This approach requires that the ovjboss process be running. If the ovjboss process is not running, an nnmsnmp*.ovpl command behaves as follows:

- For SNMPv1 and SNMPv2c agents, the command uses default values for any unspecified communication settings.
- For SNMPv3 agents, if you specify a user and password, the command uses default values for any unspecified communication settings. If you do not specify a user and password, the command fails.

Evaluate the following items and create a communication plan:

- Default communication settings
- Communication configuration regions
- Specific node configurations
- Retry and timeout values
- Active protocols
- Multiple community strings or authentication profiles

5.2.1 Default communication settings

Because NNMi uses default values to complete any configuration settings that were not specified for the applicable region or specific node, set defaults to be reasonable for the majority of your network.

- Are there commonly-used community strings that NNMi has to try?
- What default timeout and retry values are reasonable in your network?
- Which SNMP traps are sent from network devices to NNMi?

5.2.2 Communication configuration regions

Regions represent areas of the network where similar communication settings make sense. For example, the local network around the NNMi management server usually 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 areas into one region based on similar lag times. Consider the following network map:



For timeout and retry purposes, you might want to configure the following regions:

- Region A for Net 1
- Region B to include Net 10, Net 20, and Net 30

5. NNMi Communications

• Region C 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 path or the two-hop path from the NNMi management server.

Regions are also used to group devices with similar access credentials. If all routers in your network use the same community string (or a small set of possible community strings) and you can identify the routers with a naming convention (for example, rtrnnn.yourdomain.com), you can configure a region containing all routers so that they are handled similarly. If you cannot use a wildcard to group the devices, you can configure each as a specific node.

Plan your region configurations so that you can apply the same timeout and retry values and access credential configurations to all nodes in a region.

Region definitions can overlap, and a device might qualify for multiple regions. NNMi applies the settings from the region with the lowest ordering number (and no other matching regions).

5.2.3 Specific node configurations

For any device with unique communication configuration requirements, use the specific node settings to specify the communication settings for that node. Example uses of specific node settings include the following:

- A node that might not respond well to SNMPv2c/SNMPv3 GetBulk requests
- A node whose name does not match the name pattern of other similar nodes

You can enable or disable SNMP communication for a specific device. For details, see *Specific Node Settings Form* (*Communication Settings*) in NNMi Help.

5.2.4 Retry and timeout values

Configuring longer timeouts and more retries can result in more responses from devices that are busy or distant. This higher response rate 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 might require a period of testing and adjusting values in your environment.

To get an idea of the current lag time for each hop, execute the following commands:

- In Windows: Execute tracert for a device in each network area
- In Linux: Execute traceroute for a device in each network area

5.2.5 Active protocols

You can control the type of traffic NNMi generates when communicating with devices in your network by using communication and monitoring configuration settings. Use the communication settings when firewalls in your infrastructure prohibit ICMP or SNMP traffic. Use monitoring settings to fine-tune protocol usage when you do not need a particular subset of data about devices. If either communication settings or monitoring settings disable a protocol for a device, NNMi does not generate that type of traffic to the device.

^{5.} NNMi Communications

Note

Disabling SNMP communication significantly compromises device management capabilities, such as troubleshooting, because the details of devices cannot be obtained.

Note whether each region or specific device needs to receive ICMP traffic.

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.

If Web Agents are configured (in addition to SNMP Agents), NNMi can use additional protocols (for example, SOAP protocol for VMware environments).

5.2.6 Community strings and authentication profiles

Plan the community strings and authentication profiles to be tried for each area of your network. For the default and region settings, you can configure multiple community strings and authentication profiles to be tried in parallel.

Note

While trying probable community strings, NNMi queries might cause devices to generate authentication failures. Inform your operations department that authentication failures might safely be ignored while NNMi completes its initial discovery. Alternatively, you can minimize the number of authentication failures by configuring as tightly as possible your regions and the associated community strings and authentication protocols to try.

If your environment uses SNMPv1 or v2 and SNMPv3, determine the minimum acceptable security level for each region.

(1) SNMPv1 and SNMPv2 community strings

For regions where SNMPv1 or SNMPv2c access is acceptable, gather the community strings in use within the region and any unique community strings required by specific devices.

(2) SNMPv3 authentication profiles

For regions containing SNMPv3-accessible devices, determine the minimum acceptable default authentication profiles, the authentication profiles appropriate for each region, and the unique authentication credentials in use on specific devices (if any). Also determine the authentication and privacy protocols in use within your network. You can specify one authentication protocol and one privacy protocol for each specific node or region setting.

For SNMPv3 communication, NNMi supports the following authentication protocols:

- HMAC-MD5-96
- HMAC-SHA-1

For SNMPv3 communication, NNMi supports the following privacy protocols:

- DES-CBC
- TripleDES

5. NNMi Communications

- AES-128
- AES-192
- AES-256

5.3 Configuring communications

This section explains the following:

- Configuring SNMP proxy settings
- Using NETCONF for device support
- Discover and monitor VMware hypervisor-based virtual networks
- Discover and monitor Cisco ACI networks
- Multihomed NNMi Management Server

After reading the information in this section, see *Configuring Communication Protocol* in NNMi Help for specific procedures.

Note

It is a good idea to save a copy of the existing configuration before you make any major configuration changes. For details, see 4.2 Best practice: Save the existing configuration.

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.

🛛 Тір

Double-check the ordering numbers for the defined regions. If a node qualifies for membership in multiple regions, NNMi applies to that node the settings from the region with the lowest ordering number.

5.3.1 Configuring an SNMP proxy

Some networks use an SNMP proxy agent to communicate with network devices. The following diagram shows the SNMP communication steps NNMi uses if you configure an **SNMP Proxy Address** and an **SNMP Proxy Port** using **Configuration** > **Communication Configuration** from the NNMi console.

Q Тір

For the alternate method of configuring SNMP proxy settings from the command line, see the nnmcommunication.ovpl reference page.

Figure 5-1: Using proxy servers



- 1. The NNMi management server sends an SNMP request to the SNMP proxy address and SNMP proxy port to obtain information from the managed router and switch. The NNMi management server uses the special proxy varbind SecurityPackAgentAddressOid OID(.1.3.6.1.4.1.99.12.45.1.1) to encode the remote address and port of the managed router and switch, and then adds this varbind to the SNMP request.
- 2. The SNMP proxy server reads this special proxy varbind, determines the destination of the SNMP request, and then sends the SNMP request to the managed router and switch to obtain the information requested by the NNMi management server.
- 3. The managed router and switch respond to the SNMP proxy server (using the SNMP proxy address and SNMP proxy port) and return the requested information.
- 4. The SNMP proxy server responds to the NNMi management server (using the configured SNMP port). Note: NNMi supports SNMP proxy servers that support using the SecurityPackAgentAddressOid OID (.1.3.6.1.4.1.99.12.45.1.1). Use the following property to include this OID in SNMP requests for devices using SNMP proxy settings:

com.hp.nnm.snmp.USE PROXY VARBIND=true

The default setting for this property is false.

- 5. The SNMP proxy server forwards SNMP informs and traps from the managed devices to NNMi. NNMi supports the use of the following OIDs to determine the source of incoming traps forwarded from an SNMP proxy:
 - TrapForwardingAddressTypeOid.1.3.6.1.4.1.11.2.17.2.19.1.1.2.0 (HP)
 - TrapForwardingAddressOid .1.3.6.1.4.1.11.2.17.2.19.1.1.3.0 (HP)
 - Rfc3584TrapAddressOid .1.3.6.1.6.3.18.1.3.0 (RFC 3584)
 - Rfc3584TrapCommunityOid .1.3.6.1.6.3.18.1.4.0 (RFC 3584)

When using NNMi with an SNMP proxy server, ask the proxy vendor if they support the OIDs in this list.

5.3.2 Using NETCONF for device support

NNMi relies primarily on the Simple Network Management Protocol (SNMP) as the method for collecting management information from supported devices. However, NNMi might also use the Network Configuration Protocol (NETCONF) for some specific vendor devices whose required management information is not reported when SNMP is used.

```
5. NNMi Communications
```

```
JP1/Network Node Manager i Setup Guide
```

Currently, NNMi uses NETCONF to support Juniper Networks QFabric systems only.

The following sections provide a brief introduction to NETCONF and explain the configuration required for both the managed devices and NNMi in order to support such devices in NNMi.

(1) What is NETCONF?

NETCONF, like SNMP, is an Internet Engineering Task Force (IETF) standard for network management. NETCONF is defined by IETF Request for Comments (RFC) 4741 and 4742 (Version 1), later updated by RFC 6241 and 6242 (Version 1.1).

NETCONF is intended primarily for use as a device configuration mechanism, whereas SNMP is used most commonly for monitoring, polling, and fault notification. Both protocols report management information that is useful to NNMi.

NNMi uses NETCONF to collect information about devices during discovery or rediscovery (in other words, it collects read-only information). NNMi does not use NETCONF to modify the device configuration or to monitor status or provide performance metrics.

NETCONF is an XML-formatted command-and-response protocol that runs primarily over Secure Shell (SSH) transport. The NETCONF protocol is similar in some ways to the traditional device console Command Line Interface (CLI), except that the XML-formatted commands and results are designed for management applications, rather than human interaction with the devices.

Because NETCONF is a relatively new management protocol, it is not as widely available among device vendors as is SNMP.

Note the following in the case of a vendor who has implemented NETCONF in a device that NNMi is managing:

- NETCONF commands are generally more vendor specific and are not as well publicized as the many standard and vendor-specific MIBs in SNMP. Consequently, NNMi's capability to utilize NETCONF is still quite limited.
- Where a specific vendor implements NETCONF in its devices and reports the management information that NNMi needs, you must add that device-specific NETCONF support in NNMi.

For details, see (3) Enabling and configuring NETCONF in a managed device and (4) Configuring NETCONF device credentials in NNMi.

(2) NETCONF protocol operation

Details of NETCONF communication between NNMi and the managed devices are transparent to the NNMi user. However, the following overview might be helpful for troubleshooting:

1. A NETCONF client (management application, such as NNMi) establishes an SSH connection with the NETCONF server (subsystem) on a managed device.

Valid SSH user name and password credentials must be specified by the client and authenticated by the device.

- 2. The client application and device exchange capabilities in the form of <hello> messages.
- 3. The client initiates requests to the device in the form of Remote Procedure Call (RPC) messages, including standard <get> or <get-config> operations, plus any vendor-specific operations that are defined for the device.
- 4. The device responds with the results of operations in the form of RPC reply messages.
- 5. When the client application is done sending requests and processing the responses, it sends a <close-session> RPC message to the device.
- 6. The device acknowledges with an < 0k > RPC reply message.

5. NNMi Communications

7. Both sides then terminate the SSH connection.

(3) Enabling and configuring NETCONF in a managed device

Before NNMi can communicate with a managed device, it might be necessary to explicitly enable and configure NETCONF in that device. See your vendor's device configuration documentation for specific instructions.

In general, the following prerequisites must be satisfied on the managed device:

- Enable NETCONF on either the default NETCONF TCP port 830 or on the standard SSH TCP port 22.
- Configure SSH user name and password credentials on the device for NETCONF communication access.

NNMi requires only read-only access.

(4) Configuring NETCONF device credentials in NNMi

You must configure NETCONF SSH credentials in NNMi to match those configured in the managed device before NNMi can use NETCONF for communicating with that device.

If proper NETCONF credentials are not configured for a device, NNMi discovery proceeds (using SNMP only); however, the management information reported in NNMi for that device might be incomplete.

You use **Communication Configuration** in the NNMi console to configure NETCONF device credentials settings in the **Device Credentials** tab of the relevant **Specific Node Settings**, **Regions**, or **Default Device Credentials** for the device.

Important

You can configure only a single SSH user and password for each managed device. This means that the same set of credentials is used for both regular SSH and NETCONF sessions with that device.

Once configured, NNMi uses the new credentials during the next discovery cycle for the specified device (node).

For details about how to edit the NNMi Communication Configuration forms, see NNMi Help for Administer.

5.3.3 Discover and monitor VMware hypervisor-based virtual networks

(1) Prerequisites to Monitor Virtual Machines Hosted on Hypervisors

NNMi supports:

- Discovery and monitoring of supported hypervisors. On the hypervisor's node form, each virtual machine is listed on the Hosted Nodes tab.
- Discovery and monitoring of each virtual machine (routers, switches, nodes, etc.). On the virtual machine's node form, a Hosted On Node attribute shows the hypervisor's name.

The following table describes the pre-requisites for discovering hypervisors and the virtual machines

What you want to discover?	Prerequisite(s)	For more information
Hypervisor	The hypervisor must support SNMP communication and be accessible from NNMi using SNMP.	Not Applicable
	NNMi must be configured to communicate with the associated SNMP Agent (IP Address and Community String or SNMPv3 authentication).	To configure using NNMi user interface, see Help for Administer > Configuring Communication Protocol, see instructions for SNMP settings for Default, Regions, or Specific Nodes. To configure using command line interface (CLI), see the nnmcommunication.ovpl reference page, or the Linux manpage, for more information.
	NNMi must be configured to communicate with the hypervisor using HTTPS. Note: (VMware only) You must replace the VMware default certificate (localhost.localdomain) with a certificate that is generated using the hostname of the ESXi server. For more information, see the VMware documentation.	To configure using CLI, see "(3) Configuring NNMi to Communicate with Hypervisors Using HTTPS". To configure using NNMi user interface, see Help for Administer > Configuring Communication Protocol, instructions for Trusted Certificate Settings for Default, Regions, or Specific Nodes.
Virtual Machines on the hypervisor	In addition to the SNMP requirements mentioned for hypervisors, you need to configure the hypervisor device credentials in NNMi to authenticate with the hypervisor's web-service.	To configure using NNMi user interface, see Help for Administer > Configuring Communication Protocol, instructions for Credential Settings for Default, Regions, or Specific Nodes. To configure using CLI, see nnmcommunication.ovpl reference page.

(2) Replacing the VMware Default Certificate

Important

The self-signed or CA-signed certificate must be generated using the fully qualified domain name as the hostname for the ESXi server.

By default, a VMware certificate uses localhost.localdomain as the hostname for the ESXi server.

It is necessary to replace the VMware default certificate with the certificate generated by using the host name of the ESXi server. For details about the procedure, see the VMware documentation.

(3) Configuring NNMi to Communicate with Hypervisors Using HTTPS

This section provides instructions to upload certificates by using the CLI. For upload instructions using NNMi user interface, see Help for Administer > Configuring Communication Protocol.



- If you need to use HTTP to communicate with hypervisors, also see "(4) Enable HTTP to Communicate with Hypervisors".
- To enable NNMi to monitor VMs hosted on a hypervisor (such as VMWare ESXi) using HTTPS protocol, you must upload the hypervisor's trusted certificate to NNMi by using one of the following options:

- Upload trusted certificate using NNMi user interface.
- Upload trusted certificate by using command line interface (CLI).
- A Trusted Certificate is an SSL certificate that NNMi uses to establish trusted connection with hypervisors using HTTPS protocol. At Default and Region levels, it is a CA certificate that NNMi uses to trust hypervisors that use the certificates issued by the same CA. At Node level, it is the hypervisor's SSL certificate (self-signed or CA signed) generated by using FQDN as the subject name.

To upload a trusted certificate to NNMi, follow these steps:

1. Obtain the hypervisor's trusted certificate and copy it to a temporary location on the NNMi management server.



(VMware only) You must replace the VMware default certificate (localhost.localdomain) with a certificate that is generated using the hostname of the ESXi server. For more information, see the VMware documentation.

- 2. Verify that the certificate is of the supported format. The supported trusted certificate file extensions are .pem, .crt, .cer and .der.
- 3. Execute the appropriate command to upload the certificate at the required level. From the following table, choose the command that meets your requirements:

Level	Purpose	Command
Default (Global)	To upload a trusted certificate at the default level for organizations that use certificates signed by the same CA on hypervisors globally.	nnmcommunication.ovpl addCertificate -default -cert <fully certificate="" file="" path="" qualified="" the="" to=""></fully>
Region	To upload a trusted certificate for the region for organizations that use certificates signed by the same CA on hypervisors in a given region.	nnmcommunication.ovpl addCertificate -region <region name="" or="" uuid=""> -cert <fully path="" qualified="" to<br="">the certificate file></fully></region>
Node	To upload an SSL certificate (CA-Signed or Self- Signed certificate) used on a specific hypervisor. Note: The self-signed or CA-signed certificate must be generated using the fully qualified domain name (FQDN) as the subject name.	nnmcommunication.ovpl addCertificate -nodeSetting <node name="" or="" uuid=""> -cert <fully path="" qualified="" the<br="" to="">certificate file></fully></node>

Sample Commands:

- Default: nnmcommunication.ovpl addCertificate -default -cert /tmp/new.pem
- **Region:** nnmcommunication.ovpl addCertificate -region region1 -cert /tmp/ region1.der
- Node: nnmcommunication.ovpl addCertificate -nodeSetting node1 -cert /tmp/ node1.crt
- 4. Upon successful execution, the command output displays information about the uploaded certificate. Verify the certificate information.



• You can view or delete the uploaded certificates by using listCertificates and removeCertificate commands. See the nnmcommunication.ovpl reference page for more information.

• After a hypervisor is discovered, you can upload, replace, or delete a certificate directly on the Web Agent by using the command updateWebagentSettings. See the nnmcommunication.ovpl reference page for more information.

(4) Enable HTTP to Communicate with Hypervisors

By default. NNMi uses the HTTPS protocol to communicate with hypervisors.

If you need to use HTTP, add the required property to the server.properties file:

1. Navigate to the server.properties file:

Windows:%NnmDataDir%nmsas\NNM\server.properties

Linux: \$NnmDataDir/nmsas/NNM/server.properties

2. Add the following lines:

```
#Determines whether http should be used to communicate with SOAP agents such as th
e VMware vSphere API.
#HP recommends this property only be enabled in demonstration or test environments
and that HTTPS be
#configured for production environments.
nms.comm.soap.targetconfig.HTTP_ENABLED=true
```

3. Restart the NNMi management server:

Run the $\ensuremath{\mathsf{ovstop}}$ command on the NNMi management server.

Run the ovstart command on the NNMi management server.

To disable HTTP for hypervisor communication:

1. Navigate to the server.properties file:

Windows:%NnmDataDir%nmsas\NNM\server.properties

Linux:\$NnmDataDir/nmsas/NNM/server.properties

2. Change the HTTP_ENABLED property value to false:

nms.comm.soap.targetconfig.HTTP_ENABLED=false

3. Restart the NNMi management server:

Run the ovstop command on the NNMi management server.

Run the ovstart command on the NNMi management server.

4. Follow the steps described in "(3) Configuring NNMi to Communicate with Hypervisors Using HTTPS".

5.3.4 Discover and monitor Cisco ACI networks

If you want to discover and monitor networks running on Cisco ACI, you must perform the following additional tasks:

Task 1: Create a read-only user in the Cisco APIC console

Create a Cisco APIC user with read-only privilege; the user must have read access to all REST APIs of Cisco APIC. This user will be used by NNMi to discover Cisco APIC systems. While creating the user, select **all** in the **Security Domain** section on the **Create Local User** page.

Task 2: Configure NNMi to communicate with Cisco APIC systems

For each Cisco APIC system that you want to discover, provide the access credentials by using the **Device Credentials** form. These credentials help NNMi connect with Cisco APIC systems. Also, to facilitate HTTPS communication between NNMi and Cisco APIC systems, you must upload Cisco ACI or CA-trusted certificates to the NNMi management server.

To have the system recognize a Cisco ACI node as a Cisco ACI node, it is necessary to ensure that each node is detected as an SNMP node.

Follow these steps to complete this task:

1. Identify one Cisco APIC system in each cluster that you want to discover. Make sure that the SNMP agent is enabled on every Cisco APIC system in all the clusters that you want to discover.

Discovering only one Cisco APIC system in an APIC cluster enables NNMi to eventually discover all the Cisco APIC systems in the cluster.

2. Obtain all trusted certificates for use with Cisco APIC systems.

You can use a set of certificates where each certificate is specific for a particular Cisco APIC system; you can use CA-signed certificates; you can use a combination of the two.

3. Configure NNMi to communicate with Cisco APIC systems by accessing the Cisco ACI APIs.

a. Obtain the credentials that were created in Task 1: Create a read-only user in the Cisco APIC console.

b. In the Communication Configuration form, go to the Specific Node Settings tab.

🛛 Тір

You will be able to access the API of Cisco ACI and communicate with the Cisco APIC system regardless of whether you are using region settings, or are using the default settings.

c. Add a new node.

In the **Specific Node Settings** tab, click **New**, and then define a new node in the **Specific Node Settings** form.

Or, double-click an existing node.

- d. In the Specific Node Settings form, go to the Device Credentials tab.
- e. Click New.
- f. Specify the SNMP v1 or v2c community string or SNMPv3 credentials in the SNMPv1/v2 Community Strings or SNMPv3 Settings tab.
- g. In the **Specific Node Device Credentials** form, select **Type** as **CiscoACI**, and then specify the credentials of the Cisco ACI user that you created in **Task 1: Create a read-only user in the Cisco APIC console**.
- h. Upload trusted certificates. Skip this step if you want to configure HTTP communication.
 - a. In the Specific Node Settings form, go to the Trusted Certificates tab.
 - b. Click Upload Certificate.

The **Open** window appears.

c. In the Open window, select a certificate, and then click Open.

You can use any one of the following:

- A CA-signed certificate to communicate with the Cisco APIC system You can use only the following certificate formats:
 - .pem
 - .crt

5. NNMi Communications

- .cer
- .der

i. Click Save & Close.

Task 3: Configure and run discovery

Configure NNMi to seed the Cisco APIC systems in the environment. While configuring seeding, specify fully qualified domain names or IP addresses of one Cisco APIC system in each cluster that you want to discover.

Wait for NNMi discovery to gather information. NNMi discovers the APIC clusters and all the Cisco ACI Leaf and Spine nodes managed by the discovered Cisco APIC systems.

Note

If the settings described in task 1 to task 3 are not specified and Cisco ACI nodes are detected, it is necessary to run configuration polls for the Cisco APIC nodes twice in order to monitor Cisco ACI network using this function.

5.3.5 Multihomed NNMi Management Server

When an NNMi management server is configured to have multiple IP addresses, managed nodes always use the IP address configured with the operating system on the NNMi management server. If you want to configure NNMi to use a non-default IP address while communicating and exchanging data with managed nodes, follow the procedure in this section.

- 1. Log on to the NNMi Management Server.
- 2. Open the following file with a text editor:
 - Windows: %NnmDataDir%shared\nnm\conf\props\nms-communication.properties
 - Linux: /var/opt/OV/shared/nnm/conf/props/nms-communication.properties
- 3. To set a non-default IPv6 address, uncomment the com.hp.ov.nms.comm.snmp.sourceAddress.IPv6 property, and then set the property to an IPv6 address of your choice.
- 4. To set a non-default IPv4 address, uncomment the com.hp.ov.nms.comm.snmp.sourceAddress.IPv4 property, and then set the property to an IPv4 address of your choice.

Important

- If NNMi is installed in an HA cluster, you must reconfigure NNMi. The value of the NNM_INTERFACE property in the ov.conf file must match the value specified with the com.hp.ov.nms.comm.snmp.sourceAddress.IPv4 property in the nms-communication.properties file. For the location of the ov.conf file, see 19.9.1 NNMi HA configuration files.
- If you change a file while the system is operating in High Availability (HA) mode, you must make the change on both nodes in the cluster. When you use NNMi in an HA configuration, if the change requires you to stop and restart the NNMi management server, you must place the nodes in maintenance mode before running the ovstop and ovstart commands. For details, see 19.6.1 Placing NNMi in maintenance mode.

5. Save the changes.

6. Execute the following commands to restart the NNMi services:

ovstop -c ovstart -c

5.4 Evaluating communications

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

Consider the following:

- Are all nodes configured for SNMP? See 5.4.1 Checking all nodes for SNMP configuration.
- Is SNMP access currently available for a device? See 5.4.2 Checking SNMP access.
- Is the management IP address correct? See 5.4.3 Checking the management IP address for SNMP Devices.
- Is NNMi using the correct communication settings? See 5.4.4 Checking the communication settings.
- Do the State Poller settings agree with the communication settings? See 5.4.5 Checking whether the monitoring configuration matches the communication settings.

5.4.1 Checking all nodes for SNMP configuration

- 1. Open the Nodes inventory view.
- 2. Filter the Device Profile column to contain the string No SNMP.
 - For each device 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).

5.4.2 Checking SNMP access

- 1. Select the node in an inventory view.
- 2. Choose Actions, Polling, and then Status Poll, or choose Actions, Polling, and then Configuration Poll. If the results show any SNMP values, communication is operational.

You can also test communication from the command line with the nnmsnmpwalk.ovpl command. For details, see the *nnmsnmpwalk.ovpl Reference Page*.

5.4.3 Checking the management IP address for SNMP Devices

To determine which management address NNMi has selected for a device, follow these steps:

1. Select the node in an inventory view.

2. Choose Actions, Configuration Details, and then Communication Settings.

3. In the Communication Configuration window, verify that the management address of the SNMP agent listed in the **Active SNMP Agent Settings** list is correct.

5.4.4 Checking the communication settings

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

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

- 1. Select the node in an inventory view.
- 2. Choose Actions, Configuration Details, and then Communication Settings.

NNMi evaluates all region and default settings by using a specific node match and ordering number to obtain the displayed value.

3. In the Communication Configuration window, verify that the values listed in the SNMP configuration settings table are the settings you want NNMi to use for this node.

If the communication settings are not correct, use the source information in the SNMP configuration settings table as a starting point for fixing the problem. You might need to change the configuration or the ordering number of a region or specific node.

Note

For VMware communication, verify the active settings in the Web Agent form or use the nnmcommunication.ovpl listWebAgentSettings command.

For more information, see the NNMi Help for Administer.

5.4.5 Checking whether the monitoring configuration matches the communication settings

Even if the communication settings permit 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. Choose Actions, Configuration Details, and then 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.

^{5.} NNMi Communications

5.5 Tuning communications

Reducing 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.

Tuning timeouts and retries

When NNMi attempts to contact a device using SNMP during discovery, the communication configuration determines whether NNMi can 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 affect 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.

Reducing default community strings

Having a large number of default community strings can negatively affect 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.

^{5.} NNMi Communications



NNMi Discovery

One of the most important network management tasks is keeping your view of the network topology current. NNMi 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 NNMi Help. If you have experience working with NNM and you want to see how discovery has changed in NNMi, see <u>25.1</u> Network discovery for a high-level overview of the differences.

6.1 Concepts of discovery

NNMi's default setting for discovering only routers and switches enables you to focus your network management on the critical or most important devices. In other words, target the backbone of the network first. In general, 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 the devices to be discovered and included 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.

Important

NNMi does not perform any default discovery. You must configure discovery before any devices appear in the NNMi topology.

Each discovered node (physical or virtually hosted) 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.

When tracking license information, note the following:

- Consumption: NNMi discovers and manages nodes up to the NNMi licensed capacity limit (rounded up):
- VMware environments: Each device with a Device Profile of vmwareVM is equivalent to 1/10th node. All other devices are equivalent to one discovered node.

If the number of discovered nodes reaches or exceeds the licensed capacity limit, no new nodes are discovered unless one of the following occurs:

- Install a license extension.
- Review your configuration settings and limit NNMi discovery to only the important nodes in your network environment. Then, delete nodes and let NNMi rediscovery reset the managed inventory of nodes.

For details about the configuration for discovering many nodes, see NNMi Help.

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 devices beyond the limits of your responsibility. (For details about State Poller, see 7. NNMi State Polling.)

NNMi provides two primary discovery configuration models:

- *List-based discovery* -- Uses a list of seeds to explicitly tell NNMi exactly which devices to add to the database for monitoring.
- *Rule-based discovery* -- Tells NNMi which areas of your network and device types to add to the database, gives NNMi a starting address in each area, and then lets NNMi discover the defined devices.

You can use any combination of list-based and rule-based discovery to configure the devices for NNMi to 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 uses tenants to support networks containing overlapping address domains. Overlapping address domains might exist in the static network address translation (NAT), dynamic network address translation (NAT), or port address

^{6.} NNMi Discovery

JP1/Network Node Manager i Setup Guide

translation (PAT) area in the network management domain. To handle such a network, NNMi uses seeded discovery to place overlapping address domains in different tenants. For details, see NNMi Help.

Important

- If you are using NNMi to manage VMware Hypervisor-Based Virtual Networks, see the "Tenants within Virtual Environments" help topic in the Help for Administer.
- If you plan to configure multi-tenancy, configure tenants before initiating network discovery.

6.1.1 Device profiles and device attributes

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 groups and stores them in the topology portion of the database. System capabilities are visible on the **Node** form. However, these capabilities are not used by the 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.

Note

If Web Agents are configured (in addition to SNMP Agents), NNMi can use additional protocols (for example, SOAP protocol for VMware environments).

NNMi ships with a large number of 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.

Evaluate the following areas:

- Selecting your primary discovery approach
- Auto-discovery rules
- Node name resolution
- Subnet connection rules
- Discovery seeds
- Rediscovery interval
- Do-not-discover objects
- Interface discovery range

6.2.1 Selecting your primary discovery approach

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

(1) List-based discovery

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

NNMi uses tenants to support networks containing overlapping address domains. Overlapping address domains might exist in the static network address translation (NAT), dynamic network address translation (NAT), or port address translation (PAT) area in the network domain management domain. To handle such a network, NNMi uses seeded discovery to place overlapping address domains in different tenants. For details, see NNMi Help.

Important

- If you are using NNMi to manage VMware Hypervisor-Based Virtual Networks, see the "Tenants within Virtual Environments" help topic in the Help for Administer.
- If you plan to configure multi-tenancy, list-based discovery is the recommended discovery approach.

Benefits of using list-based discovery only include:

- · Provides very tight control over what NNMi manages
- Supports the specification of a non-default tenant at discovery time
- 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 list-based discovery only include:

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

(2) Rule-based discovery

With rule-based discovery, you create one or more auto-discovery rules to define the areas of the network for NNMi to 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 such device is covered by an auto-discovery rule).

Disadvantages of using rule-based discovery include:

- It is easier to confront 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 than you want to manage, you might want to delete unneeded devices from the NNMi topology and node deletion can be time consuming.
- All non-seeded nodes receive the default tenant at discovery. If you want to use NNMi multi-tenancy, you must update the tenant assignment after discovery.

6.2.2 Creating auto-discovery rules

(1) Configuring auto-discovery rules

When you configure auto-discovery rules, you specify the following:

- Auto-discovery rule ordering
- What devices to exclude from discovery
- Whether to use Ping sweep
- What discovery seeds, if any, to use

(2) 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 apply. 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 autodiscovery 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.
(3) Excluding devices from discovery

- To prevent discovery of certain object types, you can 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 such an auto-discovery rule a low ordering number, the discovery process quickly passes by the objects that satisfy 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.
- The addresses listed on the **Excluded IP Addresses** tab of the **Discovery Configuration** form apply to all autodiscovery rules. Unless they are configured as discovery seeds, these addresses are never added to the NNMi topology. (Discovery seeds are always discovered.)

Note

Some networks use routing protocols such as Hot Standby Router Protocol (HSRP) and Virtual Router Redundancy Protocol (VRRP) to provide router redundancy. When routers are configured in a router redundancy group (RRG), the routers in the RRG share a protected IP address (one active and one standby). NNMi does not support discovery and management of multiple RRGs configured with the same protected IP address. Each RRG must have a unique protected IP address.

(4) Ping sweep

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.

Note

• 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.

• Because firewalls often regard Ping sweeps as attacks on the network, a firewall might block all traffic from a device that emits Ping sweeps.

Q Тір

Enable Ping sweep for small discovery ranges only.

(5) Discovery hints from SNMP traps

NNMi processes the source IP addresses of received SNMP traps as hints to auto-discovery rules. For details about SNMP trap incidents, see NNMi Help for Administer.

(6) Discovery seeds for auto-discovery rules

Provide at least one discovery seed per auto-discovery rule. The options for providing the seeds are listed below. Specify one or a combination of these discovery seeds.

• Enter seeds on the **Discovery Seed** form by clicking **Seeds** under **Discovery** in the **Configuration** workspace.

- Use the nnmloadseeds.ovpl command to load information from a seed file.
- Enable Ping sweep for the rule, at least for initial discovery.
- Configure a device to send SNMP traps to the NNMi management server.

(7) 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 from discovery some IP addresses that are 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.*.

(8) Examples

Discovery rule overlap

Figure 6-1 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 the auto-discovery rules.

Figure 6-1: 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 (1.3.6.1.4.1.11.2.39). Leave the IP address range unset.

6.2.3 Changing the order for 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

Important

If you change a node's host name, there is a delay before NNMi data reflects the name change, because NNMi caches DNS names to enhance performance.

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 for defining the sysName for each new device as it is added to the network. In such a case, you would set sysName as the first choice for node name resolution so that NNMi can discover a 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.



- 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.
- 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.

6.2.4 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 discovery on that address.

Q Тір

Use the default connection rules. Modify them only if there is a problem.

6.2.5 Discovery seeds

This subsection explains the devices that are used as discovery seeds.



• One of the NNMi 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.

• 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 host name 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 that influences your monitoring and status calculations.

Rule-based discovery only

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. 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.

🛛 Тір

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.

6.2.6 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 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.

6.2.7 Do-not-discover objects

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

• On the **Communication Configuration** form, you can turn off ICMP communication, SNMP communication, or both at different levels: globally for communication regions, or for specific host names or IP addresses. For details about the impacts of disabling one or both of these protocols, see 5.1.7 Polling protocols.

- 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 will still appear on the map and in the database, but spiral discovery will 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 from the database specific IP address ranges, IP addresses, or both. 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.
- On the **Discovery Configuration** form, on the **Excluded IP** Addresses page, you can configure an excluded IP address filter to exclude a range of IP addresses from discovery.

If all of a node's IP addresses are entered into the **Excluded IP Addresses** list after that node has already been discovered, NNMi does not delete the node. In addition, NNMi does not delete the entire history of a node unless the NNMi administrator intentionally deletes the node from the NNMi database.

When an IP address range is excluded, any overlapping addresses in the static network address translation (NAT), dynamic network address translation (NAT), or port address translation (PAT) area in the network management domain are also excluded.

NNMi uses tenants to support networks containing overlapping address domains. To handle such a network, NNMi uses seeded discovery to place overlapping address domains in different tenants. For details, see NNMi Help.

• On the **Discovery Configuration** form, on the **Excluded Interfaces** page, you can select specific types of interfaces to exclude from the discovery process. For details, see NNMi Help.

6.2.8 Interface discovery range

NNMi enables you to define a filter to specify a range of interfaces to be discovered. This is especially useful for discovering only subsets of interfaces when the node is large. When the **Excluded Interfaces** option is used, interfaces are filtered after information has been obtained from devices. On the other hand, when an interface discovery range is specified, information about interfaces that are not in the specified range is not requested from NNMi. Therefore, range-based discovery can improve discovery performance on large devices when only some of the interfaces are managed.

On the **Included Interface Ranges** tab of the **Discovery Configuration** form you use the system object ID prefix value and ifIndex value to define an interface range. For details, see NNMi Help.

6.3 Configuring discovery

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

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

- Complete all communication settings.
- Complete all auto-discovery rules (if any).
- Configure subnet connection rules.
- Configure name resolution preferences.
- Use Save and Close to save all the configuration forms and return to the console.

Note

If you perform rule-based discovery, it is advisable to save a copy of the existing configuration before you make any major configuration changes. For details, see 4.2 Best practice: Save the existing configuration.

6.3.1 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.

6.3.2 Tips for configuring seeds

When configuring seeds, note the following best practices:

- 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 host names, DNS provides the IP address for each node.)
- Formats for the entries in the seed file are as follows:

```
IP_address # node name
IP_address2, "tenant-name-or-tenant-UUID" # node name
```

- 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 on 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 **Seeds** view in the **Discovery** area of the **Configuration** workspace on the NNMi console, and then re-enter the node on the **Discovery Seed** form on the NNMi console, or run the nnmloadseeds.ovpl command.

^{6.} NNMi Discovery

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. (When you save the information on the **Discovery Seed** form, NNMi updates the seed configuration immediately.)

6.3.3 Discovering link aggregation

Link aggregation requires an NNMi Advanced license.

A Link Aggregation (LAG) protocol enables network administrators to configure a set of interfaces on a switch as one aggregator interface. Such a configuration creates an aggregator layer 2 connection to another device using multiple interfaces in parallel to increase bandwidth, the speed at which data travels, and redundancy.

For details, search for link aggregation in NNMi Help.

6.3.4 Discovering server-to-switch link aggregation (S2SLA)

Link aggregation requires an NNMi Advanced license.

Network administrators often need additional reliability and better resource usage between servers and switches. Many network administrators choose to use the Link Aggregation Configuration Protocol (LACP) because of its widespread use by network equipment providers. LACP is negotiated automatically after the IT engineer has bonded the ports on both sides of the server-to-switch configuration.

Network administrators often choose to use one of two types of switch-to-server connections to achieve the reliability and resource usage between servers and switches that they need:

- Option 1: Bond two or more ports on the server and connect them to the same number of ports on the switch. If a port on either the server or the switch fails, a backup port is activated.
- Option 2: Bond both the server and switch to provide the aggregate total bandwidth of all the ports in the aggregation.

NNMi provides a Discovering Server-to-Switch Link Aggregations (S2SLA) feature to help you manage switch-toserver connections. To ensure that NNMi can properly discover S2SLA information for a node, complete the following tasks:

- By default, Linux does not install Net-SNMP, which is its SNMP agent package. If Net-SNMP is missing from your NNMi management server, you must install it.
- The bonding interface on Linux can assume the MAC address of one of the aggregated interfaces, but it does not have to do so. The bonded interface can have a MAC address that does not belong to any of the server's interfaces.

Note

All interfaces in the aggregation use the same MAC address. A check of the SNMP interfaces table returns the same MAC for the aggregator and aggregated interfaces. The shared MAC is used in outbound packets. The access switch's FDB table shows this MAC as being heard over the switch's aggregated interface.

To view the original MAC addresses, use the following command:

cat /proc/net/bonding/bond0

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

6.4.1 Following 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:

- On the **Database** tab of the System Information window, watch for the node count to reach the expected level and stabilize. This window does not refresh automatically. During initial discovery, open the System Information window several times.
- Under **Discovery** in the **Configuration** workspace, look at the **Seeds** tab. Refresh this tab until all seeds show the Node created result, which indicates that a 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 (located on the General tab) 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 from different areas of your network are present.
- Open the Layer 2 Neighbor 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.

6.4.2 Checking for discovery of all seeds

1. Open the Seeds view.

2. On the Seeds view, 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 host name in a hostnolookup.conf file.

If a name cannot be resolved due to the IP address, include the corresponding IP address in the ipnolookup.conf file. For details, see the reference pages for hostnolookup.conf and ipnolookup.conf.

- License node count exceeded -- This scenario occurs when the number of devices already discovered has reached your license limit. You can either delete some discovered nodes or purchase additional node pack licenses. When tracking license information, note the following:
 - Consumption: NNMi discovers and manages nodes up to the NNMi licensed capacity limit (rounded up):
 - VMware environments: Each device with a Device Profile of vmwareVM is equivalent to 1/10th node. All other devices are equivalent to one discovered node.
- Node discovered but no SNMP response -- SNMP communication problems can occur for seeded devices as well as for devices that are discovered through auto-discovery. For details, see 5.4 Evaluating communications.

6. NNMi Discovery

6.4.3 Checking for valid device profiles

- 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 (by choosing **Device Profiles** from **Configuration**), and then perform a configuration poll on the node to update its data.

6.4.4 Checking for discovery of all nodes

To avoid discovery problems, make sure that NNMi manages only nodes that use a unique IP address that does not appear for any other node in the management domain. For example, if a node suddenly disappears or gets merged with another node in the database, and it is part of a router redundancy group (RRG), there are special requirements. To manage a router that participates in an RRG, you must use a unique IP address (which is not a protected address) as the management address of the router, and SNMP must be enabled on that address. NNMi will not properly manage a router if it tries to use a protected IP address as the management address.

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 5.4.1 Checking all nodes for SNMP configuration.

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

- For 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.

6.4.5 Evaluating the auto-discovery rules (Rule-based discovery only)

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 whether to create a node. 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:

- Discover Matching Nodes 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. If **Discover Any SNMP Device** is enabled, NNMi discovers all SNMP devices. If **Discover Non-SNMP Devices** is enabled, NNMi also discovers non-SNMP devices. Enabling these settings without considering your environment can result in NNMi discovering more nodes than intended.

(1) 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

6.3.1 Tips for configuring auto-discovery rules.) 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 there is no discovery of some devices you expected to discover, 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 than 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 Device** is enabled.

(2) 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.

6.4.6 Evaluating connections and VLANs

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**, **Maps**, and then **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.

6.4.7 Rediscovering a device

To verify that the device has been deleted:

- 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.

6.5 Tuning discovery

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

- Filter by IP address range, system object ID, or both.
- Limit discovery of non-SNMP devices and any SNMP devices (devices that are not switches or routers).
- To delete one or more nodes from the NNMi database on the command line, use the nnmnodedelete.ovpl command. This command deletes nodes, but not seed definitions, from the NNMi database. To delete one or more seed definitions from the NNMi database on the command line, use the nnmseeddelete.ovpl command.
- There are special discovery circumstances that might be remedied by suppressing discovery protocol collections. For details, see 21.24 Suppressing the use of discovery protocols for specific nodes.

6.5.1 Deleting unresponsive objects

You can control the deletion of the following unresponsive objects by specifying the number of days to wait after an object has become unresponsive:

- Unresponsive nodes
- Connections that are down

To delete unresponsive nodes, perform the following steps:

- 1. In the Configuration workspace, click Discovery, and then Discovery Configuration.
- 2. In the **Delete Unresponsive Objects Control** area, enter the numbers of days for the system to wait before deleting the applicable objects.

A value of zero (0) indicates that no objects are to be deleted. After the specified waiting period, the unresponsive objects are deleted from the database.

Note

When Delete Unresponsive Nodes is enabled, NNMi does not delete virtual machine nodes under any of the following circumstances:

- The VM does not support an SNMP agent
- The VM does not have any IP addresses because VMware tools not installed
- The IP address fault monitoring for the VM is not configured

For more information, see the "Configure Whether to Delete Unresponsive Nodes" help topic in the Help for Administer.



NNMi State Polling

This chapter provides information to help you expand and fine-tune network monitoring by configuring the NNMi State Poller service. This chapter supplements the information in NNMi Help. For an introduction to how monitoring works and for detailed information about how to configure monitoring, see Monitoring Network Health in NNMi Help. If you are using NNM version 8 or earlier and you want to see how monitoring in NNMi has changed, see 25.2 Status monitoring for a high-level overview of the differences.

7.1 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 7.2 Planning state polling for more specific information.

As with network discovery, focus network monitoring on the critical or most important devices in the network. NNMi can only poll devices in the topology database. You can 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 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 internally the mapping from the type of data collection you enable to the actual MIB objects, significantly simplifying configuration.

Note

If Web Agents are configured (in addition to SNMP Agents), NNMi can use additional protocols (for example, SOAP protocol for VMware environments).

As you plan the polling configuration, 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 4.6 Node groups and interface groups and 4.7 Node/interface/ address hierarchy.

7.1.1 Order of evaluation

Because an interface or node might 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, the State Poller performs evaluation as follows:

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 the lowest order number to the highest. The first matching group is used and evaluation stops. Any contained interface that 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.

7.2 Planning 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 to capture during the polling process.

7.2.1 Polling checklist

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

□ What can NNMi monitor?

□ What are the logical groups for monitored items, based on object type, location, relative importance, or other criteria?

□ How often does NNMi need to monitor each group?

□ What data needs to be collected to capture information about the monitored item? This might include:

-- ICMP (ping) response

-- SNMP fault data

-- Additional SNMP component health data

Note

If Web Agents are configured (in addition to SNMP Agents), NNMi can use additional protocols (for example, SOAP protocol for VMware environments).

(1) Example polling configuration

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

1. What can NNMi monitor?

Because you can only monitor what has been discovered, you configure auto-discovery rules to ensure that NNMi's database contains your proxy servers. For details about configuring discovery, see 6. NNMi Discovery.

2. What are the logical groups for monitored items?

It makes sense 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 does NNMi need to monitor each group?

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

4. What data needs to be collected?

Here is where the monitoring configuration differs from other groups. For our proxy server example, you enable ICMP fault monitoring and disable SNMP fault polling monitoring. Without SNMP fault monitoring for the group, component health monitoring will not apply.

```
7. NNMi State Polling
```

5. Which SNMP traps are sent from my network devices to NNMi?

NNMi uses some SNMP traps to poll devices as the traps are received without waiting for the next polling interval. For more detailed planning information concerning these configuration choices, see the following topics:

- 7.2.2 What Can NNMi Monitor?
- 7.2.3 Stop Monitoring
- 7.2.4 Interfaces to Unmonitored Nodes
- 7.2.5 Extending Monitoring
- 7.2.6 Creating node and interface groups
- 7.2.7 Planning polling intervals
- 7.2.8 Planning the data to be collected
- 7.2.9 Deciding which SNMP traps to send to NNMi

7.2.2 What Can NNMi Monitor?

The State Poller Service monitors each discovered interface, address, and SNMP agent that is designated to be actively monitored in your management domain. State Poller can also be configured to provide Card, Chassis, Node Sensor, Physical Sensor, and Router Redundancy Group monitoring.

Note

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

If NNMi is monitoring a hypervisor network environment, it will also monitor additional objects, including the following:

- Hypervisors
- Virtual Machines (VMs) that are hosted on hypervisors
- Virtual Switches
- Uplinks (represented as interface objects)

Q Тір

- Ensure that VMware Tools is installed on your virtual machines and then use the Virtual Machines Node Group provided by NNMi to enable fault polling for the IP addresses associated with your VMs. This is a recommended practice to ensure that NNMi can identify any VM nodes where the underlying Virtual Machine has been deleted or moved to a hypervisor NNMi does not manage. For more information about enabling fault polling, see "Default Settings for Monitoring" in the NNMi Help for Administer.
- Use the Virtual Machines Node Group provided by NNMi to enable fault polling for the IP addresses associated with your Virtual Machines (VMs). This is a recommended practice to ensure that NNMi can identify any VM nodes where the underlying Virtual Machine has been deleted or moved to a hypervisor that NNMi does not manage. For more information, see "Default Settings for Monitoring" and "Configure Whether to Delete Unresponsive Nodes" in the NNMi Help for Administer.

For more information about monitoring, see the NNMi help. Also see "7.2.5 Extending Monitoring".

7.2.3 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 one or more objects 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.

Q Тір

Some of the devices, interfaces, or both 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 might want to create one or more node groups to enable you to set management modes more easily.

7.2.4 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:

• Discovering 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**. For nodes that you do not want NNMi to monitor, you will need to set the mode to **Not Managed**.



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

• Polling 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.

7.2.5 Extending 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.



- 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.

7.2.6 Creating node and interface 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).

🕽 Тір

Configure one set of node and interface groups for network monitoring. Configure a different set of node groups for network visualization through maps.

These groups are defined through the **Configuration** workspace, **Object Groups** folder, and then **Node Groups** option, or through the **Configuration** workspace, **Object Groups** folder, and then **Interface Groups** option. By default, they are 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** to save the definition.

You can set polling types and polling intervals at a node group level 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, devices, or both 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?

🛛 Тір

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 criterion. For example, you could put all your Cisco routers into OUT OF SERVICE mode while you apply an IOS upgrade.

(1) Interface groups

Based on your criteria, determine the interface groups you wish to create. Remember that interface groups are evaluated first (see 7.1 Concepts for state polling). Interface groups can reference node group membership, so to implement your plan you might end up configuring node groups before interface groups.

Preconfigured interface groups

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

- 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

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 (such as name, alias, description, speed, index, address, or other IFType attributes).
- 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.

(2) 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

A default collection of node groups is provided to simplify your configuration tasks. These node groups are based on device categories derived from the system object ID during the discovery process. The node groups provided by default include the following:

- Routers
- Networking infrastructure devices (such as switches or routers)
- 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 details, see *Node Groups As Predefined View Filters* in NNMi Help.

• Virtual Machines

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
- Host name wildcard convention
- Device Profile derivatives, such as category, vendor, and family
- MIB II sysName, sysContact, sysLocation

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.

🛛 Тір

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.

Interaction with device profiles

When each device is discovered, NNMi uses its system object ID to index it 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 might 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 Switch will receive the configured settings for the node group.

О Тір

If NNMi is managing a hypervisor network environment, you might want to create a Node Group that contains only Virtual Machines (VMs). These nodes are identified using the vmwareVM device profile. You can also use this Node Group to occasionally check for VMs that are no longer hosted on a hypervisor. After selecting this Node Group, filter by Hosted On = null to identify these VMs. You can also use this Node Group to enable fault polling for the IP addresses associated with your VMs, which is also a best practice to ensure your VMs continue to be monitored even when its associated hypervisor has been deleted.

7.2.7 Planning polling intervals

You select for each object group a polling interval that NNMi uses to collect data. The interval can be as short as one minute or as long as days, as best matches your service level agreements.

🖌 Тір

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

Note

The Causal Engine performs a status poll of each node every 24 hours and updates status, conclusion, and incident information as needed. This status poll does not affect the timing of the polling interval configured for the device.

7.2.8 Planning the data to be collected

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

ICMP (ping)

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

SNMP Polling

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.

Web Polling

If Web Agents are configured (in addition to SNMP Agents), NNMi can use additional protocols. For example, SOAP protocol for VMware environments.

SNMP component health data

You might 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.

Q Тір

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

7.2.9 Deciding which SNMP traps to send to NNMi

NNMi uses the following SNMP traps to poll devices when these SNMP traps are received rather than waiting for the next polling interval:

- CempMemBufferNotify
- CiscoColdStart
- CiscoEnvMonFanNotification
- CiscoEnvMonFanStatusChangeNotif
- CiscoEnvMonRedundantSupplyNotification
- CiscoEnvMonSuppStatusChangeNotif
- CiscoEnvMonTemperatureNotification
- CiscoEnvMonTempStatusChangeNotif
- CiscoEnvMonVoltageNotification
- CiscoEnvMonVoltStatusChangeNotif
- CiscoFRUInserted
- CiscoFRURemoved
- CiscoLinkDown
- CiscoLinkUp
- CiscoModuleDown
- CiscoModuleUp
- CiscoModuleStatusChange
- CiscoRFProgressionNotif
- CiscoRFSwactNotif
- CiscoWarmStart
- HSRPStateChange
- IetfVrrpStateChange
- Rc2kTemperature
- RcAggLinkDown
- RcAggLinkUp
- RcChasFanDown
- RcChasFanUp
- RcChasPowerSupplyDown
- RcChasPowerSupplyUp
- Rcn2kTemperature
- RcnAggLinkDown
- RcnAggLinkUp
- RcnChasFanDown
- RcnChasFanUp
- RcnChasPowerSupplyDown
- RcnChasPowerSupplyUp

7. NNMi State Polling

- RcnSmltIstLinkDown
- RcnSmltIstLinkUp
- RcSmltIstLinkUp
- RcVrrpStateChange
- SNMPColdStart
- SNMPLinkDown
- SNMPLinkUp
- SNMPWarmStart

To force NNMi to poll devices when these traps are received, configure your network devices to send these traps to NNMi.

Note

For details about these SNMP trap incident configurations, from the NNMi console, navigate to the **Configuration** workspace and select **Incidents** > **SNMP Trap Configuration**.

Also see (5) Discovery hints from SNMP traps.

7.3 Configuring state polling

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

Note

It is a good idea to save a copy of the existing configuration before you make any major configuration changes. For details, see 4.2 Best practice: Save the existing configuration.

7.3.1 Configuring the interface groups and node groups to be monitored

For details about the reasons for using node groups and interface groups for polling, see 7.2.6 Creating node and interface groups in the previous section.

You can use the NNMi console or CSV files to create node groups and interface groups. For example, if node group information is located on a Microsoft Excel worksheet, save this information as a CSV file, and then use the nnmloadnodegroups.ovpl command to add the file to NNMi. Similarly, you can add interface group information to NNMi by using the nnmloadinterfacegroups.ovpl command. For details, see the reference pages for nnmloadnodegroups.ovpl and nnmloadinterfacegroups.ovpl.

To create groups of nodes and interfaces on the NNMi console, use the **Configuration** workspace. For details, see *Creating Groups of Nodes or Interfaces* in NNMi Help.

Examples:

To configure a node group for ProximiT proxy servers:

- 1. From Configuration, open Object Groups > Node Groups, and then click New.
- 2. Name the group Proxy Servers and select the Add to View Filter List check box.
- 3. On the Additional Filters tab, select the hostname attribute, and set the operator to like.
- 4. For the node's host name, enter prox*.yourdomain.com, and then click Save and Close.

For value, enter the wildcard as prox*.example.com.

If you had configured a device profile and device 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.

7.3.2 Configuring interface monitoring

State Poller analyzes interface group membership before node groups. For each of the interface groups you created, as well as any pre-existing ones you want to use, open the **Monitoring Configuration** form and the **Interface Group**

Settings tab to create a custom set of instructions for how you want State Poller to handle that group. Your instructions will include the following:

- Enabling or disabling fault monitoring
- Setting the fault polling interval
- Selecting whether NNMi will monitor unconnected interfaces (or unconnected interfaces hosting IP addresses) in the group

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.

🖌 Тір

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.

7.3.3 Configuring 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 for how you want State Poller to handle that group. Your instructions can include the following:

- Enabling or disabling fault monitoring
- Setting the fault polling interval
- Selecting whether NNMi will monitor unconnected interfaces (or unconnected interfaces hosting IP addresses) in the group

You can configure different settings for each interface group.

🛛 Тір

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.

7.3.4 Specifying the default settings for monitoring

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.

Note

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

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

7.4.1 Verifying 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.

To verify the configuration for network monitoring, use the following checks:

- (1) Is the interface or node a member of the right group?
- (2) Which settings are being applied?
- (3) Which data is being collected?

(1) 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 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 might 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 **Node Groups** or **Interface 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 by choosing Configuration, and then 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 might need to choose **Actions**, **Polling**, and then **Configuration Poll** to update the attributes for the node so that it qualifies.

(2) 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**, **Configuration Details**, then **Monitoring Settings**. NNMi opens the current monitoring settings.

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

You might need to check the settings for the object to ensure traffic has not been disabled for it; you do this by choosing **Actions**, **Configuration Details**, and then **Communication Settings**.

^{7.} NNMi State Polling

(3) 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.

Q Тір

If Web Agents are configured (in addition to SNMP Agents), NNMi can use additional protocols.

For example, SOAP protocol for VMware environments.

Select a node, and then click **Actions** > **Polling** > **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.

7.4.2 Evaluating 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.

State Poller health information tells you whether the Status Poller is able to keep up with polling requests.

(1) Is the State Poller keeping up?

At any time, you can check the current health statistics about the State Poller service on the **State Poller** tab of the System Information window, as described in the following table.

Information	Description
Status	Overall status of the State Poller service
Poll counters	 Collections requested in last five minutes Collections completed in last five minutes Collections in process Collection request delays
Time to execute skips in last five minutes	 The number of regularly scheduled polls that did not complete within the configured polling interval. A non-zero value indicates that the polling engine is not keeping up or that targets are being polled faster than they can respond. What to watch for: If this value continues to increase, there are problems communicating with the target or NNMi is overloaded. Action to take: Look in the nnm.log file for messages for the classes beginning with the string com.hp.ov.nms.statepoller to determine the targets for the skipped polls. If the skipped polls are for the same targets, change the configuration to poll these targets at a less frequent rate or to increase the timeout for these targets. If the skipped polls are for different targets, check the NNMi system performance, especially the available memory for ovjboss.
Stale collections in last five minutes	 A stale collection is a collection that has not received a response from the polling engine for at least 10 minutes. A healthy system will never have any stale collections. What to watch for: If this value increases consistently, there is a problem with the polling engine. Action to take: Look in the nnm.log file for messages for the classes beginning with the string com.hp.ov.nms.statepoller to determine the targets for the stale collections. If the stale collections are for a single target, unmanage the target until you can resolve the problem.

Table 7-1: State Poller health information

Information	Description
Stale collections in last five minutes	If the stale collections are for different targets, check the performance of the NNMi system and the NNMi database. Stop and restart NNMi.
Poller result queue length	 What to watch for: Makes sure that this value is close to 0 most of the time. If it is greater than 0, take the following action: Action to take: If this queue size is very large, ovjboss might be running out of memory.
State mapper queue duration	 What to watch for: Make sure that this value is close to 0 most of the time. If it is greater than 0, take the following action: Action to take: If this queue size is very large, check the performance of the NNMi system and the NNMi database.
State updater queue time	 What to watch for: Make sure that this value is close to 0 most of the time. If it is greater than 0, take the following action: Action to take: If this queue size is very large, check the performance of the NNMi system and the NNMi database.
State updater exceptions	What to watch for: This value should be 0.

7.5 Tuning state polling

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

- The number of devices and 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. To maintain the most specific control and to select the smallest subset of interfaces to poll, enable these pollings on the **Node Settings** or **Interface Settings** form (not in **Default Settings** on the **Monitoring Configuration** form).
- 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 might 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 component health monitoring has no effect on the timeliness of polling. It simply gathers additional MIB objects at the schedule time. However, disabling component health monitoring might reduce the amount of memory used by the State Poller.



NNMi Incidents

NNMi provides a large number of default incidents and correlations. The default incidents enable you to display incidents immediately on the NNMi console. The correlations enable you to reduce the number of incidents to be managed. This chapter provides information to help you fine-tune network management by configuring NNMi incidents. This chapter supplements the information in NNMi Help. For an introduction to NNMi incidents and for detailed information about how to configure incidents, see Configuring Incidents in NNMi Help. If you have experience working with NNM version 8 or earlier and you want to see how event monitoring has changed in NNMi, see 25.3 Customizing event monitoring.

8.1 Concepts for incidents

NNMi collects network status information from the following sources:

- The NNMi Causal Engine, which analyzes the health of your network and provides an ongoing health status reading for each device. The Causal Engine also extensively evaluates and determines the root cause of network problems whenever possible.
- SNMP traps from network devices. The NNMi Causal Engine uses this information as symptoms during its analysis.

NNMi converts this network status information into incidents that provide useful information for managing the network. NNMi provides many default incident correlations that reduce the number of incidents for network operators to consider.

You can customize the default incident correlations and create new incident correlations to match the network management needs of your environment.

The incident configurations on the NNMi console define the incident types that NNMi can create. If no incident configuration matches a received SNMP trap, that information is discarded. NNMi always discards an incoming trap when the management mode of the source object is set to **Not Managed** or **Out of Service** in the NNMi database or the device is not monitored by fault polling.

nnmtrapconfig.ovpl-dumpBlockList outputs information about the current incident configuration, including SNMP traps that were not passed into the incident pipeline because of nonexistent or disabled incident configurations.

Additionally, NNMi discards SNMP traps from network devices that are not in the NNMi topology. For details about changing this default behavior, see *Handle Unresolved Incoming Traps* in NNMi Help.

For details, see How NNMi Gathers Incidents in NNMi Help.

8.1.1 Incident lifecycle

The following table describes the stages of an incident's lifecycle.

Table 8-1:	NNMi	incident	lifecycle
------------	------	----------	-----------

Lifecycle state	le state Description		Incident used by	
None	The NNMi event pipeline receives input from all sources and creates incidents as needed.	Not applicable	• NNMi	
Dampened	The incident is in a holding place waiting to be correlated with another incident. The purpose of this waiting period is incident reduction in the incident viewers. The dampening interval can vary by incident type. For details, see 8.1.7 Incident suppression, enrichment, and dampening.	NNMi	• NNMi	
Registered	The incident is visible in incident views. The incident is forwarded to any configured destinations (Northbound or global manager).	NNMi A user can also set this state in an incident view.	UsersLifecycle transition actions	

Lifecycle state Description		State set by	Incident used by		
In progress	The incident has been assigned to someone who is investigating the problem. The network administrator defines the specific meaning of this state.	User	UsersLifecycle transition actions		
Completed	Investigation of the problem indicated by the incident is complete and a solution is in place. The network administrator defines the specific meaning of this state.	User	UsersLifecycle transition actions		
Closed	NNMi has determined that the problem reported by this incident no longer constitutes a problem. For example, when you remove an interface from a device, all incidents related to that interface are automatically closed.	User or NNMi	UsersLifecycle transition actions		

8.1.2 Trap and incident forwarding

The following table summarizes the ways to forward traps and incidents from the NNMi management server to another destination.

Item	NNMi trap forwarding	NNMi Northbound interface trap forwarding	Global network management trap forwarding	
What to forward	SNMP traps from network devices	SNMP traps from network devicesNNMi management events	SNMP traps from network devices	
Forwarding format	SNMPv1, SNMPv2c, or SNMPv3 traps, as received (SNMPv3 traps can be converted to SNMPv2c traps)	SNMPv2c traps created from NNMi incidents	NNMi incidents	
Added information	In most cases, NNMi adds varbinds to identify the original source object. NNMi does not ever modify SNMPv1 traps.	NNMi adds varbinds to identify the original source object.	Any information added to the incident by the regional manager processes is retained in the forwarded incident.	
Where to configure	Incidents > Trap Server > Trap Forwarding Configuration in the Configuration workspace	Northbound Interface in the Integration Module Configuration workspace	Forward to Global Managers tab on an SNMP Trap Configuration form or syslog configuration.	
Notes		Forward the remote incident are visible in the global methods i		
For details	<i>Configure Trap Forwarding</i> in NNMi Help		Configure Forward to Global Manager Settings for an SNMP Trap Incident (NNMi Advanced) in NNMi Help	

Table 8-2: Supported ways to forward traps and NNMi incidents

Legend:

--: Not applicable

8.1.3 Received SNMP traps

You can use either of the following approaches to forward to another application the SNMP traps that NNMi receives from managed devices:

- Use the NNMi SNMP trap forwarding mechanism.
- Use the NNMi Northbound interface SNMP trap forwarding mechanism.

The approach to trap identification by the receiving application varies as follows:

• Windows (all) and Linux (without original trap forwarding)

This description applies to the default and to the SNMPv3-to-SNMPv2c conversion forwarding options.

The NNMi SNMP trap forwarding mechanism on a Windows NNMi management server enriches each SNMP trap before forwarding it to the trap destination. The trap appears to originate from the NNMi management server. (This information also applies to a Linux NNMi management server for which the original trap forwarding option is not selected on the **Trap Forwarding Destination** form.)

To ensure the correct association between the trap-sending device and the event in the receiving application, the rules for these traps must be customized for the enriched varbinds. Interpret the value from the originIPAddress (.1.3.6.1.4.1.11.2.17.2.19.1.1.3) varbind. The originIPAddress value is a byte string of generic type InetAddress, either InetAddressIPv4 or InetAddressIPv6, as determined by the value of the originIPAddressType (.1.3.6.1.4.1.11.2.17.2.19.1.1.2) varbind. The rule must read the originIPAddressType varbind to determine the type of Internet address value (ipv4(1) or ipv6(2)) in the originIPAddress value to a display string.

For details about the varbinds that NNMi adds to forwarded traps, see *Trap Varbinds Provided by NNMi* in NNMi Help, RFC 2851, and the following file:

- Windows: %NNM_SNMP_MIBS%\Vendor\Hewlett-Packard\hp-nnmi.mib

- Linux: \$NNM_SNMP_MIBS/Vendor/Hewlett-Packard/hp-nnmi.mib

• Linux with original trap forwarding

A Linux NNMi management server can forward traps in the same format in which NNMi receives them. Each trap appears as if the managed device sent it directly to the trap destination, so existing trap processing configured in the receiving application will work without modification.

• NNMi Northbound interface (all operating systems)

The NNMi Northbound interface enriches each SNMP trap before forwarding it to the trap destination. The trap appears to originate from the NNMi management server. To ensure the correct association between the trap-sending device and the event in the receiving application, the rules for these traps must be customized for the enriched varbinds.

The nnmiIncidentSourceNodeHostname (1.3.6.1.4.1.11.2.17.19.2.2.21) and nnmiIncidentSourceNodeMgmtAddr (1.3.6.1.4.1.11.2.17.19.2.2.24) varbinds identify the original source object.

8.1.4 MIBs

NNMi requires that the following management information base (MIB) files be loaded into the NNMi database:

- All MIB variables used in MIB expressions for the Custom Poller feature, line graphs, or both
- Sensors that NNMi monitors for health (for example, fan or power supply)

NNMi requires that the management information base (MIB) files, or the traps defined in the MIB files, be loaded into the NNMi database.

8.1.5 Custom incident attributes

NNMi uses custom incident attributes (CIAs) to attach additional information to incidents.

- For an SNMP trap incident, NNMi stores the original trap varbinds as CIAs for the incident.
- For a management event incident, NNMi adds pertinent information (for example, com.hp.ov.nms.apa.symptom) as CIAs for the incident.

You can use incident CIAs to narrow the scope of configurations such as incident lifecycle transition actions, suppression, deduplication, and enrichment. You can also use CIAs to narrow the availability of the application menu items for an incident view or form.

To determine the CIAs that NNMi adds for any given incident, open a sample incident from an incident view, and look at the information on the **Custom Attributes** tab.

(1) CIAs added to closed management event incidents

When the NNMi Causal Engine determines that the conditions that caused a management event incident no longer apply, NNMi sets that incident's lifecycle state to CLOSED and adds the CIAs listed in the table below to the incident. NNMi console users can see this information in the **Correlation Notes** field of the **Incident** form. Lifecycle transition actions can use the values of the CIAs directly.

Name	Description				
cia.reasonClosed	The reason that NNMi canceled or closed the incident. This reason is also the conclusion name, such as NodeUp or InterfaceUp. When this field is not set, it means that an NNMi console user closed the incident. To determine the NNMi expected values of a cia.reasonClosed CIA, see <i>How NNMi Closes Incidents</i> in NNMi Help.				
cia.incidentDurationMs	The duration, in milliseconds, of the outage, as measured by NNMi from when the status goes down to when it comes back up. This value is the difference between the cia.timeIncidentDetectedMs and cia.timeIncidentResolvedMs CIAs. It is a more accurate measurement than comparing the down and up incident timestamps.				
cia.timeIncidentDetectedMs	The timestamp, to the millisecond, when the NNMi Causal Engine first detected the problem.				
cia.timeIncidentResolvedMs	The timestamp, to the millisecond, when the NNMi Causal Engine detected that the problem has been resolved.				

	Table 8-3:	Custom	incident	attributes	for a	closed	incident
--	------------	--------	----------	------------	-------	--------	----------

NNMi adds the CIAs listed in Table 8-3 to most primary and secondary root cause incidents. For example, a NodeDown incident can have InterfaceDown and AddressNotResponding incidents as secondary root causes. When NNMi closes the NodeDown incident, NNMi also closes the secondary incidents and adds the CIAs with values for each incident context to the secondary incidents.

NNMi does not add the CIAs listed in Table 8-3 to the following default management event incident types:

- Incidents that an NNMi console user closes manually
- · Incidents that NNMi closes in response to an object being deleted from the NNMi database
- IslandGroupDown incidents
- NnmClusterFailover, NnmClusterLostStandby, NnmClusterStartup, and NnmClusterTransfer incidents
- Incidents in the following families:
 - Correlation
 - License
 - NNMi health
 - Trap analysis

8.1.6 Incident reduction

NNMi provides the following customizable correlations for reducing the number of incidents that network operators see on the NNMi console:

• Pairwise correlation

Logically related incidents that need not be displayed in the **Incident** view (such as CiscoLinkUp following CiscoLinkDown) are managed as related incidents. Specifically, when an interface changes from LinkDown to LinkUp, the corresponding LinkDown and LinkUp messages are suppressed.

• Deduplication correlation

When multiple copies of an incident are received within the specified time window, correlate the duplicates under a deduplication incident. The time window restarts for each newly received duplicate incident. In this way, NNMi correlates the duplicate incidents until it has not received any duplicates for the entire duration of the correlation time window.

Rate correlation

When the specified number of copies on an incident are received within the specified time window, correlate the duplications under a rate incident. NNMi generates the rate incident when the specified number of incidents has been received, regardless of how much time remains in the time window.

8.1.7 Incident suppression, enrichment, and dampening

NNMi provides a rich feature set for deriving the most value from incidents. For each incident type, you can use the following incident configuration options to define specifically whether an incident is of interest:

• Suppression

An incident that matches the suppression configuration does not appear in the NNMi console incident views. Incident suppression is useful for incidents (such as SNMPLinkDown traps) that are important for some nodes (routers and switches, for example) but not for others.
• Enrichment

When an incident matches the enrichment configuration, NNMi changes one or more incident values (for example, severity or message) according to the contents of the incident. Incident enrichment is useful for processing traps (for example, RMONFallingAlarm) that carry the distinguishing information in the trap varbinds (payload).

• Dampening

When an incident matches the dampening configuration, NNMi delays activity (such as refreshing the display of the incident view or execution of an action) for that incident for the duration of the dampening interval. Incident dampening provides time for the NNMi Causal Engine to perform root cause analysis on the incident, which is useful for providing fewer, more meaningful incidents on the NNMi console.

For each incident type, NNMi provides the following levels of configuration for suppression, enrichment, and dampening:

• Interface group settings

Specify incident behavior when the source object is a member of an NNMi interface group. You can specify different behavior for each interface group.

• Node group settings

Specify incident behavior when the source object is a member of an NNMi node group. You can specify different behavior for each node group.

• Default settings

Specify default incident behavior.

The following is the procedure for determining the behavior of a specific incident for any incident configuration area (suppression, enrichment, or dampening):

- 1. Check the interface group settings:
 - If the source object matches any interface group settings, carry out the behavior defined in the match with the lowest ordering number and stop looking for a match.
 - If the source object does not match any interface group settings, go to step 2.
- 2. Check the node group settings:
 - If the source object matches any node group settings, carry out the behavior defined in the match with the lowest ordering number and stop looking for a match.
 - If the source object does not match any node group settings, go to step 3.
- 3. Perform the actions defined in the default settings, if any.

8.1.8 Lifecycle transition actions

A lifecycle transition action is an administrator-provided command that runs when an incident lifecycle state changes to match the action configuration. An incident action configuration is specific to one lifecycle state for one incident type. The action configuration identifies the command to run when this incident type transitions to the specified lifecycle state. The command can include arguments that pass incident information to the action code.

The action code can be any Jython file, script, or executable that runs correctly on the NNMi management server. The action code can be specific to one incident type or it can process many incident types. For example, you might create action code that pages a network operator when NNMi creates a ConnectionDown, NodeDown, or NodeOrConnectionDown incident. You would configure three incident actions, one for the **Registered** lifecycle state for each of these incident types. Similarly, the action code can be specific to one lifecycle state change or it can respond to several lifecycle state changes. For example, you might create action code that generates a trouble ticket when NNMi creates an InterfaceDown incident and closes the trouble ticket when the InterfaceDown incident is canceled. You would configure two incident actions for the InterfaceDown incident, one for the **Registered** state and one for the **Closed** state.

Each action configuration can include a payload filter based on CIAs that limits when the action is to run. For additional filtering, you can use incident enrichment to add a CIA to the incident. NNMi determines the value of that attribute from the incident source. For example, if you have added a custom attribute to some nodes, you can add this information to the incident as a CIA and then base the payload filter for an incident action on this attribute value.

8.2 Planning incidents

Make decisions in the following areas:

- SNMP traps to be processed
- Incidents to be displayed
- How NNMi responds to incidents

8.2.1 Planning the SNMP traps to be processed

Identify the device traps that are of interest in your network and plan an incident configuration for each trap. NNMi can process traps without the MIB being loaded into NNMi.

NNMi's nnmincidentcfg.ovpl -loadTraps script enables you to automate creation and updating of SNMP trap incident configurations by using MIB files. If the MIB file contains TRAP-TYPE or NOTIFICATION-TYPE macros, you can obtain the information required for incident configurations.

Decide whether you want to see traps from devices that are not in the NNMi topology.

8.2.2 Planning the incidents to be displayed

The default set of incidents is a good place to start. You can expand and reduce the incident set over time.

Plan which incidents can be reduced though deduplication, rate configuration, and pairwise correlation.

For details, see NNMi Help for Administer.

8.2.3 Planning how NNMi responds to incidents

Plan a lifecycle status that determines the actions to be taken by NNMi when certain incidents occur. For example, an action might be sending a message via email to a network operator.

For details, see NNMi Help for Administer.

8.3 Configuring incidents

For details about how to configure incidents, see Configuring Incidents in NNMi Help.

Note

It is a good idea to save a copy of the existing configuration before you make any major configuration changes. For details, see 4.2 Best practice: Save the existing configuration.

8.3.1 Configuring incident suppression, enrichment, and dampening

Note the following about configuring incident suppression, enrichment, and dampening:

- For each interface group, node group, or default setting, you can specify a payload filter that further refines when the configuration is applicable.
- Configure interface group settings on the Interface Settings tab of an incident configuration form.
- Configure node group settings on the Node Settings tab of an incident configuration form.
- Configure default settings on the **Suppression**, **Enrichment**, and **Dampening** tabs of an incident configuration form.

8.3.2 Configuring lifecycle transition actions

Note the following about configuring lifecycle transition actions:

• By default, NNMi runs actions at the following location:

```
Windows:
```

```
%NnmDataDir%shared\nnm\actions
```

Linux:

```
$NnmDataDir/shared/nnm/actions
```

If an action is not at this location, specify the absolute path to the action in the **Command** field of the **Lifecycle Transition** Action form.

Important

Jython files must be placed in the actions directory.

- Each time you make a change to the action configuration, NNMi rereads the actions directory for Jython files and loads them into NNMi.
- Actions are enabled as a group for an incident type.
- For details about the NNMi information that you can pass to an action, see *Valid Parameters for Configuring Incident Actions (SNMP Trap Incident)* in NNMi Help.

8.3.3 Configuring trap logs

NNMi can record all incoming SNMP traps in a log file (text or CSV file). The traps are recorded at the following location:

- Windows: %NnmDataDir%log\nnm
- Linux: \$NnmDataDir/log/nnm

Use the nnmtrapconfig.ovpl script to configure the trap log file. You can select the following formats:

- CSV (default): Traps are recorded in CSV format (trap.csv).
- LOG: Traps are recorded in text format (trap.log).
- BOTH: Traps are recorded in both CSV and text formats (two log files).
- OFF: Traps are not recorded.

For example, to record traps in the BOTH mode, use the following command:

nnmtrapconfig.ovpl -setProp trapLoggingMode BOTH -persist

When the -persist argument is used, all trap server properties remain enabled, even after the trap service has been restarted. When the -persist argument is not used, all trap server properties are enabled only until the service is stopped.

Traps are written to a rolling log file. When the log file reaches the defined maximum size (defined by the nnmtrapconfig.ovpl script), the file is renamed to trap.*format*.old, and a new file replaces the existing file.

For details, see the nnmtrapconfig.ovpl Reference Page. Also see Configure Trap Logging in NNMi Help.

8.3.4 Configuring incident logs

You can configure incident logs so that incoming incident information will be written in the incident.csv file. This feature is useful for tracking and archiving incident logs.

To configure and enable incident logs, navigate to **Incident Logging Configuration** in the **Incident Configuration** area in the **Configuration** workspace. For details, see NNMi Help.

8.3.5 Configuring trap server properties

Use the nnmtrapconfig.ovpl script to configure trap server properties (nnmtrapserver.properties).

You must use the nnmtrapconfig.ovpl script to edit the nnmtrapserver.properties file. You must not edit this file directly.

The trap server properties are set to the following default values.

JP1/Network Node Manager i Setup Guide

Table 8-4:	Trap server	properties and	their default values

Trap server property	Default value
com.hp.ov.nms.trapd.udpPort	162
com.hp.ov.nms.trapd.rmiPort	1,097
com.hp.ov.nms.trapd.trapInterface	All interfaces
com.hp.ov.nms.trapd.recvSocketBufSize	53,248 kilobytes
com.hp.ov.nms.trapd.pipeline.qSize	50,000 traps
com.hp.ov.nms.trapd.connectToWinSNMP	false
com.hp.ov.nms.trapd.blocking	true
com.hp.ov.nms.trapd.blockTrapRate	50 traps/second
com.hp.nms.trapd.unblockTrapRate	50 traps/second
com.hp.ov.nms.trapd.overallBlockTrapRate	150 traps/second
com.hp.nms.trapd.overallUnblockTrapRate	150 traps/second
com.hp.ov.nms.trapd.analysis.minTrapCount	100 traps
com.hp.ov.nms.trapd.analysis.numSources	10 sources
com.hp.ov.nms.trapd.analysis.windowSize	300 seconds (5 minutes)
com.hp.nms.trapd.updateSourcesPeriod	30 seconds
com.hp.nms.trapd.notifySourcesPeriod	300 seconds
com.hp.ov.nms.trapd.hosted.object.trapstorm.enabled	false
com.hp.ov.nms.trapd.hosted.object.trapstorm.threshold	10 traps/second
com.hp.ov.nms.trapd.database.fileSize	100 megabytes
com.hp.ov.nms.trapd.database.fileCount	5 files
com.hp.ov.nms.trapd.database.qSize	300,000 traps
com.hp.ov.nms.trapd.discohint.cacheSize	5,000 entries
com.hp.ov.nms.trapd.discohint.cacheEntryTimeout	3,600 millisecond

For details, see the *nnmtrapconfig.ovpl Reference Page*.

8.4 Batch loading incident configuration

You can use the two scripts nnmincidentcfgdump.ovpl and nnmincidentcfgload.ovpl in conjunction with batch loading of incident configuration.

8.4.1 Using nnmincidentcfgdump.ovpl to create an incident configuration file

In NNMi, you can use the nnmincidentcfgdump.ovpl script to create or update an incident configuration and then use the nnmincidentcfgload.ovpl script to load the incident configuration into the NNMi database. The files are created in a non-XML format.

You can edit the files by referencing the explanation of formats stored in the following directory:

- Windows: %NnmInstallDir%examples\nnm\incidentcfg
- Linux: /opt/OV/examples/nnm/incidentcfg

To create an incident configuration file, use the following sample syntax:

```
nnmincidentcfgdump.ovpl -dump file-name -uuid -u NNMi-admin-user-name -p NNMi-admin-p assword
```

For details, see the nnmincidentcfgdump.ovpl Reference Page.

8.4.2 Using nnmincidentcfgload.ovpl to load the incident configuration

In NNMi, you can use the nnmincidentcfgload.ovpl script to load the incident configuration from a formatted configuration file into the NNMi database.

For details about the required format, see the following directory:

- Windows: %NnmInstallDir%examples\nnm\incidentcfg
- Linux: /opt/OV/examples/nnm/incidentcfg

To validate the incident configuration file before loading it into the NNMi database, use the following sample syntax:

```
nnmincidentcfgload.ovpl -validate file-name -u NNMi-admin-user-name -p NNMi-admin-pas sword
```

To load the incident configuration, use the following sample syntax:

```
nnmincidentcfgload.ovpl -load file-name -u NNMi-admin-user-name -p NNMi-admin-passwor d
```

Note the following:

• NNMi updates all settings that have a matching name or similar key identifier.

Use the nnmincidentcfgdump.ovpl script to create a non-XML file from the existing incident configuration file. If necessary, you can then edit this file before you load it into the NNMi database.

NNMi also overwrites the code values (such as incident families) associated with these settings.

- NNMi adds all incident settings that have key identifiers that are not in the NNMi database.
- NNMi does not change any existing incident setting that does not have a matching key identifier in the export file.
- NNMi resolves any unique object ID (UUID) if it is not provided in the configuration file.
- If NNMi cannot resolve a UUID, that UUID is created.

Important

If you change a file while the system is operating in High Availability (HA) mode, you must make the change on both nodes in the cluster. When you use NNMi in an HA configuration, if the change requires you to stop and restart the NNMi management server, you must place the nodes in maintenance mode before running the ovstop and ovstart commands. For details, see 19.6.1 Placing NNMi in maintenance mode.

For details, see the reference page for the nnmincidentcfgload.ovpl command.

8.5 Evaluating incidents

This section lists ways to evaluate the incident configuration:

• Verify that NNMi receives traps from all managed devices in the network. If NNMi is not receiving traps, check the firewall configuration on the NNMi management server.



Some anti-virus software includes a firewall that is configured separately from the system firewall.

- Verify that the most important traps are converted to incidents.
- Verify that incident actions run at the correct lifecycle state transitions.
- Verify that NNMi is handling incidents as expected.

The **Incident Configuration Reports** menu under **Actions** contains several options for testing an existing incident against the current configuration of that incident type. Using one of these menu items does not change the incidents currently on the NNMi console.

8.6 Tuning incidents

You can use any of the following methods to reduce the number of incidents in the NNMi console incident views:

- Disable the incident configuration for any incident types that are not needed on the NNMi console.
- Set the management mode of the network objects that you do not need to monitor to **Not Managed** or **Out of Service**. NNMi discards all incoming traps from these nodes and their interfaces.
- Set NNMi to not monitor some network objects. NNMi discards all incoming traps from source objects that are not monitored.
- Identify additional criteria for or relationships between incoming incidents. When these criteria or relationships occur, NNMi modifies the flow of incidents by recognizing the criteria or patterns of incoming management events or SNMP traps and nesting related incidents as correlated children.

8.6.1 Enabling incidents for undefined traps

By default, NNMi drops SNMP traps with no incident definition.

To generate SNMP traps with no incident definition as UndefinedSNMPTrap incidents:

- 1. Open the following file with a text editor:
 - Windows: %NNM_PROPS%\nms-jboss.properties
 - Linux: \$NNM PROPS/nms-jboss.properties
- 2. Look for the following line:

#!com.hp.nnm.events.allowUndefinedTraps=false

Edit this line as follows:

com.hp.nnm.events.allowUndefinedTraps=true

3. (Optional) Specify an incident severity.

Look for the following line:

```
#!com.hp.nnm.events.undefinedTrapsSeverity=NORMAL
```

Substitute a defined severity value for YourSpecifiedSeverity:

com.hp.nnm.events.undefinedTrapsSeverity=YourSpecifiedSeverity

The valid values are NORMAL, WARNING, MINOR, MAJOR, and CRITICAL.

4. (Optional) Specify the incident nature.

Look for the following line:

#!com.hp.nnm.events.undefinedTrapsNature=INFO

Substitute an incident nature for YourSpecifiedNature:

com.hp.nnm.events.undefinedTrapsNature=YourSpecifiedNature

The valid values are ROOTCAUSE, SECONDARYROOTCAUSE, SYMPTOM, SERVICEIMPACT, NONE, and INFO.

5. Save the changes.

```
8. NNMi Incidents
```

6. Execute the following commands to restart NNMi:

```
ovstop
ovstart
```

7. Review the list of UndefinedSNMPTrap incidents.

You must specify incident definitions for those SNMP traps that you want to display as incidents. For details, see NNMi Help.

8.6.2 Interpreting and displaying the MIB data for SNMP traps correctly

NNMi does not always know which character set to use to interpret the MIB data for SNMP traps.

The result is that NNMi displays garbled strings from MIB data (such as sysDescription and sysContact) for SNMP traps.

The solution is use the following procedure to specify the correct character set to be used by NNMi to interpret MIB data strings:

1. Open the file with a text editor:

- Windows: %NNM_PROPS%\nms-jboss.properties
- Linux: \$NNM_PROPS/nms-jboss.properties
- 2. Locate the following line and delete the comment symbols (# !):

#!com.hp.nnm.sourceEncoding=

3. Edit the com.hp.nnm.sourceEncoding property.

Add to the com.hp.nnm.sourceEncoding property the character sets supported in your environment as a comma-separated list, using the examples shown in the nms-jboss.properties file.

- 4. Save the changes.
- 5. Execute the following commands to restart NNMi:

```
ovstop
ovstart
```

If NNMi is to display MIB data in hexadecimal without using a character set, do the following:

1. Open the following file with a text editor:

- Windows: %NnmDataDir%shared\nnm\conf\nnmvbnosrcenc.conf
- Linux: \$NnmDataDir/shared/nnm/conf/nnmvbnosrcenc.conf
- 2. Add a combination of traps OID and VarBind OID.

Add a combination of traps OID and VarBind OID for the target MIB data using the examples shown in the nnmvbnosrcenc.conf file.

NNMi displays specified MIB data in hexadecimal by using the custom attribute values in the incident form.

- 3. Save the changes.
- 4. Execute the following commands to restart NNMi:

ovstop			
ovstart			



NNMi Console

This chapter explains how to use the NNMi console to configure the functions of NNMi.

9.1 A practical example of using node groups

This section explains how to configure node groups by way of a practical example.

Node groups to be configured:

 ${\tt My Network: A \ top \ level \ container \ node \ group \ containing \ other \ node \ groups}$

USA: An intermediate container node group containing other node groups

Colorado: A node group containing nodes located in Colorado

In this example, Colorado is the only node group that contains nodes.

Note the following:

- It is a best practice to design your node group map layout ahead of time.
- It is a best practice to configure one set of node and interface groups for network monitoring. Configure a different set of node groups for network visualization through maps.
- NNMi provides more than one way to configure node groups and node group maps. After you become familiar with the steps described in this document, you might find more efficient ways to create subsequent node groups and node group maps.

This section guides you through the following steps for configuring node groups and node group maps.

Creating node groups

- Step 1: Create the My Network node group.
- Step 2: Create the USA node group.
- Step 3: Create the Colorado node group using filters.
- Step 4: View the node group members to check the node group filter results.
- Step 5: Set up the node group hierarchy for the My Network node group.
- Step 6: Establish the node group hierarchy for the USA Node group.

Parent node groups might not contain any nodes, and instead they might contain only child node groups in the definition. In this example, the My Network and USA node groups are parent node groups that contain only child node groups.

Configuring the node group maps

- Step 1: Create the node group maps.
- Step 2: View the node group maps.
- Step 3: Configure node group status.
- Step 4: Configure node group map ordering.
- Step 5: Add a background image to a node group map.

9.1.1 Creating node groups

We begin by creating the node groups and then we add them to the node group maps.

(1) Step 1: Creating the My Network node group

To create the My Network node group:

- 1. Navigate to the **Configuration** workspace.
- 2. From Object Groups, select Node Groups.
- 3. Click the New icon.
- 4. In the Name attribute, enter My Network.
- 5. In the Notes attribute, enter This is the top level Node Group.
- 6. Click Save and Close to save this configuration.

(2) Step 2: Creating the USA node group

- 1. Navigate to the **Configuration** workspace.
- 2. From Object Groups, select Node Groups.
- 3. Click the New icon.
- 4. In the **Name** attribute, enter USA.
- 5. Click Save and Close to save this configuration.

(3) Step 3: Create the Colorado node group using filters.

To create the Colorado node group, use the Filter Editor to establish a filter to select the nodes.

Note

When possible, use the **Additional Filters** tab rather than specifying a list of nodes using the **Additional Nodes** tab. Using a node group filter enables NNMi to automatically place a node into the correct node group as new nodes are added to the network.

- 1. Navigate to the **Configuration** workspace.
- 2. From Object Groups, select Node Groups.
- 3. Click the New icon.
- 4. In the Name attribute, enter Colorado.
- 5. Choose the Additional Filters tab.
- 6. Click **OR** to specify that you want NNMi to match a node if the node matches either of the host name values you enter.
- 7. In the Filter Editor's Attribute field, select hostname.
- 8. Selecting **hostname** specifies that NNMi match the host name values when determining whether a node belongs to this node group.
- 9. In the **Operator** field, select like.

Selecting like enables you to use wildcard characters in the search.

- 10. In the Value field, enter a value that represents the devices you want the node group to contain. For example, cisco*.ntc.example.com represents devices named ciscovalue.network-domain.
- 11. Click Append.

9. NNMi Console

- 12. In the Attribute field, select hostname.
- 13. In the **Operator** field, select like.
- 14. In the Value field, enter a wildcard that represents the remaining device names you want to add to the Colorado node group.

For this example, use cisco?*.

- 15. Click Append.
- 16. Click Save to save the node group without closing the window.

(4) Step 4: Checking the node group filter results

To test the node group filter, you can view the members of the node group you just created.

From the Actions menu, choose Node Group Details, and then Show Members to launch a view containing all of the nodes in the node group.



Examine the node group filter definition results until you are confident the node group filter is correct.

(5) Step 5: Setting up the node group hierarchy for the My Network node group

Establish a hierarchy for the node groups, starting with the top level node group, My Network.

- 1. In the **Configuration** workspace, select **Object Groups**, and then the **Node Groups** option to view a list of the node groups you created.
- 2. Navigate to the My Network node group, and then click Open.
- 3. Click the Child Node Groups tab.
- 4. Click the New icon.
- 5. In the Child Node Group attribute, click the Lookup icon and select Quick Find.

Important

Use Quick Find to select an object, such as a node group, when it already exists.

- 6. Select USA as the child node group.
- 7. Click OK.
- 8. Click Save and Close to save your changes and close the Node Group Hierarchy form.
- 9. Click Save and Close to save your changes and close the Node Group form.

(6) Step 6: Establishing the node group hierarchy for the USA Node group

Establish Colorado as a child node group of the USA node group. Repeat the same steps described in (5) Step 5: Setting up the node group hierarchy for the My Network node group to make the Colorado node group a child of the USA Node Group.

You are ready to create the node group maps for the node groups that you created.

9.1.2 Configuring the node group maps

Follow the steps in this subsection to configure a node group map for a node group that has been created.

(1) Step 1: Creating the node group maps

To create a node group map for a node group, use the Actions menu.

1. Open the node group for which you want to create a map:

a. In the **Configuration** workspace, select **Object Groups**, and then the **Node Groups** option to view a list of the node groups you created.

b. Navigate to the node group you want and click the Open icon.

- 2. From the Actions menu, choose Maps, and then Node Group Map to display a node group map.
- 3. Position the nodes and node group map icons.
- 4. Click the Save Map icon to create the node group map.

Note

Always use **Save Map** to create a node group map, even if you have not changed any of the node positions. **Save Map** creates the node group map.

A dialog box appears for confirming that you are satisfied with the created node group map.

5. Click OK.

6. Repeat steps 1 through 5 for each node group you created.

(2) Step 2: Viewing the node group maps

To view the node group maps:

- 1. Navigate to the **Topology Maps** workspace.
- 2. Select Node Group Overview.
- 3. Select the top level map My Network.
- 4. Navigate to a child node group map by double-clicking its icon.
- 5. Use the hierarchy link at the top of the map to return to the previous map.

(3) Step 3: Configuring node group status

NNMi enables you to configure how status is calculated for a node group. When you configure node group status, you determine which of the following methods NNMi is to use:

- Use the most severe status applicable to the nodes in the node group.
- Use a calculation based on a specified percentage.

Note

Status Configuration is a global configuration. By default, NNMi uses the most severe status among the nodes in the node group.

1. Navigate to the Configuration workspace.

- 2. Select Status Configuration.
- 3. Examine the Status Configuration form to become familiar with the default percentages.

If you wish to use percentages, you must clear the **Propagate Most Severe Status** check box, and then save your changes.

(4) Step 4: Configuring node group map ordering

Node group map ordering is used to help determine the order in which maps open under the **Topology Maps** workspace. In this example, use node group map ordering to specify that the My Network node group map appears first in the list in the **Topology Maps** workspace.

- 1. Navigate to the **Configuration** workspace.
- 2. Select User Interface, and then Node Group Map Settings.

Note

As shown in the following example, the default **Topology Maps Ordering** value is 50 for all userdefined maps.

To instruct NNMi to list My Network as the first map under the **Topology Maps** workspace, change the **Topology Maps Ordering** value to a number that is less than the **Topology Maps Ordering** value for any other maps in the list (for example 5).

- 3. Open the My Network node group map.
- 4. In the Topology Maps Ordering attribute, change the value to 5.
- 5. Click Save and Close to save your changes and close the form.

You can also specify whether the map is to be displayed initially on the NNMi console. To do so, use the **User Interface Configuration** option from the **Configuration** workspace.

- 1. Navigate to the Configuration workspace.
- 2. From User Interface, click User Interface Configuration.
- 3. In the Initial View attribute, use the drop-down menu to select the workspace First Node Group in Quick Access Maps folder.

This will make the My Network map the initial view. To verify the initial view, sign out of NNMi and sign back in. The My Network map will be the view you see on the NNMi console.

(5) Step 5: Adding a background image to a node group map

To include a background graphic with a map, use the Node Group Map Settings form for the selected node group map.

- 1. Navigate to the **Configuration** workspace.
- 2. Click User Interface.
- 3. Click Node Group Map Settings.
- 4. Open the My Network node group map.
- 5. Navigate to the **Background Image** tab.
- Click http://MACHINE:PORT/nnmbg/. NNMi opens a list of graphics.

- 7. Right-click the world.png.
- 8. Copy the link location.
- 9. Close the directory listing window.

Note

Paste the copied link into the **Background Image** attribute.

Make note of the Background Image Scale value in case you want to change it later.

- 10. Click Save and Close to save your changes.
- 11. Navigate to **Quick Access Maps** in the workspace **Topology Maps** and select **My Network** to view your new map with the background graphic.

9.1.3 Deleting node groups

To delete the created Colorado node group:

- 1. Navigate to the **Configuration** workspace.
- 2. From Object Groups, click Node Groups.
- 3. From the list, select the Colorado node group, and then click the **Open** button. The Colorado node group is selected and its contents are displayed.
- 4. Click the **Delete Node Group** button.

A dialog box appears that warns the user that if the node group is deleted, all objects contained in that node group and its references will be deleted.

5. Click **OK** to delete the node group.

9.2 Reducing the maximum number of nodes displayed in a Network Overview map

The **Network Overview** map opens a map containing up to 250 of the most highly connected nodes in the Layer 3 network. If this map contains too many nodes, it might respond slowly when moving nodes or it might become too complex for practical viewing. You can change the maximum number of nodes displayed in the **Network Overview** map as shown in the following example.

Example: Suppose you want to reduce the maximum number of nodes displayed in the **Network Overview** map from 250 to 100.

To do this, follow these steps:

- 1. Edit the following file:
 - Windows: %NNM_PROPS%\nms-ui.properties
 - Linux: \$NNM_PROPS/nms-ui.properties
- 2. Look for the following line:

#!com.hp.nnm.ui.networkOverviewMaxNodes=250

Specify the maximum value for the displayed node as follows:

com.hp.nnm.ui.networkOverviewMaxNodes=100

Important

Make sure to remove the #! characters at the beginning of the line.

- 3. Save your changes.
- 4. Execute the following commands to restart NNMi:

```
ovstop
ovstart
```

9.3 Reducing the number of displayed nodes on a node group map

If you configure a node group map so that it contains hundreds of nodes, the map for that node group might show many small node icons instead of the detailed node icons you expect. To view the map with better detail, you would need to use the zoom feature. Using the zoom feature might slow the NNMi console performance when displaying maps.

The remedy is to limit the number of displayed nodes, displayed end points, or both, by doing the following:

- 1. On the NNMi console, click Configuration.
- 2. Click User Interface Configuration located beneath User Interface.
- 3. Select the Default Map Settings tab.
- 4. Modify the value shown in the Maximum Number of Displayed Nodes field.
- 5. Modify the value shown in the Maximum Number of Displayed End Points field.
- 6. Click Save and Close.

For details, see Define Default Map Settings in NNMi Help.

9.4 Configuring Gauges in the Analysis Pane

The **Gauges** tab in the analysis pane shows real-time SNMP gauges that display State Poller and Custom Poller SNMP data. These gauges display data for nodes, interfaces, custom node collections, and for node components of type CPU, Memory, Buffers, or Backplane.

You can configure the gauges by editing the following properties file:

- Windows: %NNM PROPS%\nms-ui.properties
- Linux: \$NNM_PROPS/nms-ui.properties

For each property that you want to set, if present, be sure to remove the comment characters (# !) located at the beginning of the line.

Note

The properties discussed in the sections that follow apply to ALL nodes (in other words, it is not possible to apply the properties to separate Node Groups).

😱 Tip

Make a backup copy of the nms-ui.properties file before making any changes. Be sure to place the backup copy in a directory other than the directory containing the properties file you are editing. See also the comments within the nms-ui.properties file for more information.

9.4.1 Disabling the Analysis pane

To disable the Analysis pane from the NNMi console:

1. Edit the following file:

- Windows: %NNM_PROPS%\nms-ui.properties
- Linux: \$NNM PROPS/nms-ui.properties
- 2. Locate the line that contains the following property:

```
#!com.hp.nnm.ui.analysisPaneDisabled = true
```

To disable the Analysis pane, delete #! at the beginning of the line, as shown below:

com.hp.nnm.ui.analysisPaneDisabled = true

- 3. Save the changes.
- 4. Restart NNMi by executing the following commands:

```
ovstop
ovstart
```

9.4.2 Limiting the Number of Gauges Displayed

Set the maximum number of gauges to be displayed by editing the following line and providing the desired value:

```
com.hp.nnm.ui.maxGaugePerAnalysisPanel =
```

Q Тір

A higher number of gauges affects performance when the analysis pane is displayed. A fewer number of gauges results in larger size gauges.

9.4.3 Setting the Refresh Rate for Gauges in the Analysis Pane

Set the refresh interval (in seconds) for gauges displayed in the analysis pane by editing the following property value:

com.hp.nnm.ui.analysisGaugeRefreshSecs =

🛛 Тір

Setting the value to "0" results in gauges never refreshing. A refresh rate faster than 10 seconds causes some SNMP agents to cache their values for short periods of time, causing repeated results.

9.4.4 Eliminating Gauges from the Display

Define the gauges that you do NOT want displayed (for all gauge views) by editing the following line and providing a list of gauges to eliminate from the display:

```
com.hp.nnm.ui.analysisGaugeNoDisplayKeyPatterns =
```

Note the following:

- Remove the comment character from all related lines
- You cannot have comments within a list of gauges
- Ensure that no blank lines exist within the list of gauges A blank line terminates the entries at the location of the blank line
- The default settings for this property are those in the comments These settings must be included if this configuration is being extended or amended; otherwise, an unexpected amount of gauges will appear.

9.4.5 Controlling the Order of Displayed Node Gauges

To control the order in which node gauges are displayed, edit the following line:

Note the following:

- · Wildcards are not supported in this property setting
- · Ensure that the list does not contain comments or empty lines
- The default settings for this property appear as comments. These settings must be included if this configuration is being extended or amended; otherwise, the order will not match what you configured.

9.4.6 Controlling the Order of Displayed Interface Gauges

To control the order in which interface gauges are displayed, edit the following line:

```
com.hp.nnm.ui.analysisGaugeInterfaceKeys =
```

Note the following:

- Wildcards are not supported in this property setting.
- Ensure that the list does not contain comments or empty lines.
- The default settings for this property are those in the comments. These settings must be included if this configuration is being extended or amended; otherwise, the order will not match what was anticipated.

9.4.7 Controlling the Order of Displayed Custom Poller Gauges

To control the order in which Custom Poller gauges are displayed, edit the following line:

```
com.hp.ov.nnm.ui.analysisGaugeCustomPolledInstanceKeys =
```

Note

There is no default setting for this attribute.

9.4.8 Understanding how Gauge Properties are Applied

Gauge properties are applied in the following order:

- 1. The list of all possible gauges is retrieved from State Poller.
- 2. The analysisGaugeNoDisplayKeyPatterns is first applied to remove the specified gauges from the list.
- 3. The analysisGaugeNodeComponentKeys, analysisGaugeInterfaceKeys, or analysisGaugeCustomPolledInstanceKeys is applied, as appropriate, to order the list of displayed gauges.
- 4. Finally the maxGaugePerAnalysisPanel is applied to truncate the displayed list.

9.4.9 Troubleshooting Gauge Problems

This section includes information for troubleshooting the following gauge problem:

• (1) Too Many Gauges Are Displayed

(1) Too Many Gauges Are Displayed

If you have too many gauges, do one of the following:

- Limit the number of gauges displayed using the maxGaugePerAnalysisPanel property See 9.4.2 Limiting the Number of Gauges Displayed for more information.
- Use the analysisGaugeNoDisplayKeyPatterns property to remove the gauges that are not wanted See 9.4.4 Eliminating Gauges from the Display for more information.

9.5 Configuring Map Label Scale Size and Borders

The NNMi Administrator can make the following adjustments to a map view using the nms-ui. properties file:

- The scale value for node and port labels as a map is re-sized using the Zoom feature.
- The largest relative scale factor that can be used to determine the difference in size between nodes or ports and their labels on a map.
- Whether labels for nodes and ports are surrounded with a black rectangle.

Note

By default labels for nodes and ports are surrounded with a black rectangle to improve readability when labels overlap.

The following table describes the properties to change.

Q Тір

Each scale adjustment property value is multiplied with the actual scale factors used by NNMi. For example, if you change the labelScaleAdjust value to .50, then the labels as seen on the map are one half of their normal size.

Table 9-1: Properties to Change in the nms-ui.properties File

Property	Default Value	Description
com.hp.nnm.ui.labelScaleAdjust	1.0	Adjusts the scale size of the map labels for nodes and ports
com.hp.nnm.ui.omitLabelRectangle	true	Determines whether to use a black rectangle to surround the node and port labels. Note: To turn rectangles off, set the value to false.

Note

Ë

To implement your changes, re-open or change your map view.

9.6 Configuring Auto-Collapse Thresholds for Loom and Wheel Diagrams

As NNMi administrator, you can configure the point at which Loom and Wheel diagrams initially auto-collapse

Nodes (hiding the interfaces) and Switches (hiding the ports) for better readability if the diagram is sufficiently complex. You can achieve this by adjusting the following properties in the nms-ui.properties file.

Table 9-2: Auto-Collapse Thresholds for Wheel and Loom

Property	Description	
com.hp.nnm.ui.wheelAutoCollapseThreshol d	Use this property to specify the number of labels required around the perimeter before the Wheel Diagram automatically collapses.	
com.hp.nnm.ui.loomAutoCollapseThreshold	Use this property to specify the number of labels required throughout the diagram before the Loom Diagram automatically collapses.	

To configure auto-collapse thresholds, follow these steps:

- 1. Edit the following file:
 - Windows: %NNM_PROPS\nms-ui.properties
 - Linux: \$NNM_PROPS/nms-ui.properties
- 2. Uncomment the required property, if required. See the comments in the nms-ui.properties file for details.
- 3. Update the threshold value as required and save your changes.
- 4. Reopen the diagram in NNMi console to implement your changes.

9.7 Customizing device profile icons

In NNMi, you can customize the icons that are associated with device profiles or specific nodes. These icons are displayed in table views and menu items. They are also displayed as foreground images in the NNMi topology map.

You can edit icons from the Icons option located in the User Interface folder in the Configuration workspace.

To edit or delete icons from the command line, use the nnmicons.ovpl command. For details, see the *nnmicons.ovpl Reference Page* or NNMi Help.

9.8 Overriding the refresh rate of table views

NNMi enables the NNMi administrator to override from the NNMi console the default refresh rate of table views.

The recommended minimum refresh rate is 30 seconds. Performance might be affected adversely if the refresh rate is set to less than 30 seconds.

To override the default refresh rate of NNMi table views:

1. Check the viewInfoId parameter in the URL of the view whose refresh rate is to be changed.

a. Open the view whose refresh rate is to be changed.

b. Click Show View in New Window.

c. Make a note of the value of the viewInfoId parameter in the URL.

Example:

viewInfoId=allIncidentsTableView

- 2. Edit the following file:
 - Windows: %NNM_PROPS%\nms-ui.properties
 - Linux: \$NNM_PROPS/nms-ui.properties
- 3. Add to nms-ui.properties a line in the following format for specifying the view and its refresh rate in seconds:

com.hp.ov.nms.ui.refreshViewSecs.VIEWKEYWORD = SECS

Important

- VIEWKEYWORD is the viewInfoId parameter in the view's URL.
- SECS is the refresh rate (seconds).
- Make sure that there are no extra spaces at the end of the command line.

For example, to change the refresh rate of the **All Incidents** view, to 120 seconds, add the following line to nms-ui.properties:

com.hp.ov.nms.ui.refreshViewSecs.allIncidentsTableView = 120

4. Save the changes.

5. To check the new refresh rate, open a different view, and then return to the view whose refresh rate was changed.

10

Working with Certificates for NNMi

A certificate identifies the Web server to the browser. This certificate can be self-signed or signed by a CA (Certificate Authority). The nnm-key.p12 file stores private keys and certificates with their corresponding public keys. The nnm-trust.p12 file contains certificates from other parties with whom you expect to communicate or certificates from Certificate Authorities that you trust to identify other parties. NNMi includes a self-signed certificate in both of these files (nnm-key.p12 and nnmtrust.p12). To use certain NNMi features, NNMi management servers need to share their certificates with one another. This chapter contains configuration instructions for copying these certificates among NNMi management servers and using the nnmcertmerge.ovpl script to merge these certificates into the nnm-key.p12 and nnm-trust.p12 files.

Caution

NNMi 11-50 or later version introduce a Public Key Cryptography Standards (PKCS) #12 repository to store certificates. The new PKCS #12 file-based certificate management technique is available for use as soon as you install a new instance of NNMi 11-50 or later version on a system. Environments upgraded from an older version of NNMi continue to use a JKS repository to store certificates.

In upgraded environments, you can migrate to the PKCS #12 repository by using the steps in 10.2 Configuring an Upgraded NNMi Environment to Use the New Keystore.

This section describes useful terminology to help you work with certificates.

Table 10-1: Certificate Terminology

Concept	Description
Keystore and Truststore	Truststore : NNMi truststore is the file in which you store public keys from sources that you want NNMi to trust. In a newly installed instance of NNMi 11-50 or later version, the name of the truststore file is nnm-trust.pl2.
	Caution
	On a management server where NNMi was upgraded to 11-50 or later version from an older version, the truststore file name is nnm.truststore. You can, however, perform additional steps (described in 10.2 Configuring an Upgraded NNMi Environment to Use the New Keystore) to migrate the nnm.truststore file to the nnm-trust.p12 file.
	Keystore : NNMi keystore is the file in which you import NNMi server's private key.
	In a newly installed instance of NNMi 11-50 or later version, the name of the keystore file is nnm-key.p12.
	Caution
	On a management server where NNMi was upgraded to 11-50 or later version from an older version, the keystore file name is nnm.keystore. You can, however, perform additional steps (described in 10.2 Configuring an Upgraded NNMi Environment to Use the New Keystore) to migrate the nnm.keystore file to the nnm-key.p12 file.
	These files are located at:
	• Windows: %NNM_DATA%\shared\nnm\certificates\
	• Linux: \$NNM_DATA/shared/nnm/certificates/
Default NNMi certificates	NNMi is installed with a self-signed certificate generated using default properties. You can replace the default certificate with another self-signed or CA-signed certificate.
Tools	Certificates are generated and managed using the nnmkeytool.ovpl utility (which uses Java's Keytool utility). Additionally, NNMi provides the nnmmergecert.ovpl utility to merge certificates to establish trust within NNMi systems. This program is used in HA, Failover, and Global Network Management setups.
Supported encryption algorithms	NNMi accepts certificates generated using RSA algorithm. DSA algorithm is not supported.
Self-Signed Certificate	A Self-Signed certificate is typically used for establishing secure communication between your server and a known group of clients. NNMi installs with a self-signed certificate generated using default properties.
	Note: NNMi instances configured to use a self-signed certificate will display a warning message when users try to access NNMi web console in a web browser.

Concept	Description
CA-Signed	Signed server certificate that you receive in response to the Certificate Signing Request will contain the NNMi certificate that is CA signed and one or more CA certificates (if there is more than one CA certificate, this is also known as the certificate chain).
Certificate	Note: These certificates might be in a single file or in a two separate files.
Root CA Certificate	Identifies the certificate authority that is trusted to sign certificates for servers and users.
Intermediate CA	A certificate signed by either a root or intermediate CA that is itself an authority, rather than a server or user.
Certificate	Note: The list of certificates from the NNMi server certificate to the root CA certificate, including any intermediate CA certificates, is known as the certificate chain.

^{10.} Working with Certificates for NNMi

10.2 Configuring an Upgraded NNMi Environment to Use the New Keystore

Prior to the version 11-50, NNMi used to provide a Java KeyStore (JKS) repository to store certificates. NNMi 11-50 or later version introduce a Public Key Cryptography Standards (PKCS) #12 repository to store certificates. The new PKCS #12 file-based certificate management technique is available for use as soon as you install a new instance of NNMi 11-50 or later version on a system.

However, when you upgrade an older version of NNMi to 11-50 or later version, the PKCS #12 file-based certificate management does not immediately come into effect and NNMi continues to use the JKS repository for certificate management.

With additional configuration tasks, you can configure the upgraded NNMi management server to use the new technique of PKCS #12 file-based certificate management.

To configure the upgraded NNMi management server to use PKCS #12 file-based certificate management:

- 1. Log on to the NNMi management server as root or administrator.
- 2. Run the following command to migrate to the new keystore file:

Windows:

```
%NnmInstallDir%bin\nnmkeytool.ovpl -importkeystore -srckeystore
%NnmDataDir%shared\nnm\certificates\nnm.keystore -destkeystore
%NnmDataDir%shared\nnm\certificates\nnm-key.p12 -srcstoretype JKS -deststoretype P
KCS12 -srcprovidername SUN -destprovidername SunJSSE -alias <src_alias>
```

Linux:

```
/opt/OV/bin/nnmkeytool.ovpl -importkeystore -srckeystore
/var/opt/OV/shared/nnm/certificates/nnm.keystore -destkeystore
/var/opt/OV/shared/nnm/certificates/nnm-key.p12 -srcstoretype JKS -deststoretype P
KCS12 -srcprovidername SUN -destprovidername SunJSSE -alias <src alias>
```

Caution

After running the command, you will be asked to input a password three times, as follows:

"Enter the password of the output destination keystore", "Re-enter the new password", "Enter the password of the source keystore". Enter nnmkeypass in response to all of these requests.

The new certificate management technique enables you to retain only a single certificate in the keystore at a time. In this instance, <src_alias> is the alias of the certificate in the old keystore file that you want to migrate. For the certificate alias included in the previous keystore file, specify the setting value com.hp.ov.nms.ssl.KEY_ALIAS set in the following file.

- Windows: %NNM_CONF%\nnm\props\nms-local.properties
- Linux: \$NNM CONF/nnm/props/nms-local.properties

Note

In Application Failover Environmens, <src_alias> is the alias of the certificate of the server executing the command.

3. Run the following command to migrate to the new truststore file:

Windows:

```
%NnmInstallDir%bin\nnmkeytool.ovpl -importkeystore -srckeystore
%NnmDataDir%shared\nnm\certificates\nnm.truststore -destkeystore
%NnmDataDir%shared\nnm\certificates\nnm-trust.pl2 -srcstoretype JKS -deststoretype
PKCS12 -srcprovidername SUN -destprovidername SunJSSE
```

Linux:

```
/opt/OV/bin/nnmkeytool.ovpl -importkeystore -srckeystore
/var/opt/OV/shared/nnm/certificates/nnm.truststore -destkeystore
/var/opt/OV/shared/nnm/certificates/nnm-trust.p12 -srcstoretype JKS -deststoretype
PKCS12 -srcprovidername SUN -destprovidername SunJSSE
```

Caution

After running the command, you will be asked to input a password three times, as follows:

"Enter the password of the output destination keystore", "Re-enter the new password", "Enter the password of the source keystore". Enter ovpass in response to all of these requests.

- 4. Open the server.properties file from the following location with a text editor:
 - Windows: %NnmDataDir%nmsas\nms
 - Linux: /var/opt/OV/nmsas/nms
- 5. Delete the existing content of the file.
- 6. Add the following content to the file:

```
nmsas.server.security.keystore.type=PKCS12
nmsas.server.security.keystore.file=${com.hp.ov.DataDir}/shared/nnm/certificates/n
nm-key.p12
nmsas.server.security.keystore.cred=nnmkeypass
nmsas.server.security.truststore.file=${com.hp.ov.DataDir}/shared/nnm/certificates
/nnm-trust.p12
nmsas.server.security.truststore.cred=ovpass
nmsas.server.security.keystore.alias=
nms.comm.soap.https.PROTOCOLS=TLSv1.2
```

- 7. Save the file.
- 8. Open the nms-local.properties file from the following location with a text editor:
 - Windows: %NnmDataDir%conf\nnm\props
 - Linux: /var/opt/OV/conf/nnm/props
- 9. Modify the values of all the javax parameters:

Parameter	Value
javax.net.ssl.trustStore	\${NnmDataDir}/shared/nnm/certificates/nnm-trust.p12
javax.net.ssl.trustStoreType	PKCS12
javax.net.ssl.keyStore	\${NnmDataDir}/shared/nnm/certificates/nnm-key.p12
javax.net.ssl.keyStoreType	PKCS12

10. Save the file.

- 11. Delete the nnm.keystore and nnm.truststore files from the following directory.
 - Windows: %NnmDataDir%shared\nnm\certificates

• Linux: /var/opt/OV/shared/nnm/certificates

12. Restart NNMi.

^{10.} Working with Certificates for NNMi

10.3 Using Certificates with the PKCS #12 Repository

Prior to the version 11-50, NNMi used to provide a Java KeyStore (JKS) repository to store certificates. NNMi 11-50 or later version introduce a Public Key Cryptography Standards (PKCS) #12 repository to store certificates. The new PKCS #12 file-based certificate management technique is available for use as soon as you install a new instance of NNMi 11-50 or later version on a system.

However, when you upgrade an older version of NNMi to 11-50 or later version, the PKCS #12 file-based certificate management does not immediately come into effect and NNMi continues to use the JKS repository for certificate management.

This section provides you with the procedures to work with certificates in a new installation of NNMi or an environment where the certificate repository is migrated to the PKCS #12 format.

10.3.1 Generating a Self-Signed Certificate

Caution

NNMi 11-50 or later version introduce a Public Key Cryptography Standards (PKCS) #12 repository to store certificates. The new PKCS #12 file-based certificate management technique is available for use as soon as you install a new instance of NNMi 11-50 or later version on a system. Environments upgraded from an older version of NNMi continue to use a JKS repository to store certificates.

In upgraded environments, you can migrate to the PKCS #12 repository by using the steps in 10.2 Configuring an Upgraded NNMi Environment to Use the New Keystore.

To generate a self-signed certificate, follow these steps:

- 1. Change to the directory on the NNMi management server that contains the nnm-key.pl2 and nnm-trust.pl2 files:
 - Windows: %NnmDataDir%shared\nnm\certificates
 - Linux: \$NnmDataDir/shared/nnm/certificates
- 2. Save a backup copy of the nnm-key.p12 file.
- 3. Delete the existing nnm-key.p12 file.
- 4. Generate a private key from your system.

Use the nnmkeytool.ovpl command to generate this private key:

- a. Run the following command exactly as shown:
 - Windows:

```
%NnmInstallDir%bin\nnmkeytool.ovpl -genkeypair -validity 36500 -keyalg rsa -k
eystore nnm-key.p12 -storetype PKCS12 -storepass nnmkeypass -alias <alias_nam
e>
```

• Linux:

```
$NnmInstallDir/bin/nnmkeytool.ovpl -genkeypair -validity 36500 -keyalg rsa -k
eystore nnm-key.pl2 -storetype PKCS12 -storepass nnmkeypass -alias <alias_nam
e>
```

Note

The alias, referred to as <alias_name> in this example, identifies this newly-created key. Although the alias can be any string, we recommends you use the fully-qualified domain name (FQDN) followed by a suffix to help you easily identify the right version. For example, you can use alias name as myserver.mydomain-<number> or myserver.mydomain-<date>.

b. Enter the requested information.

Note

When prompted for your first and last name, enter the FQDN of your system.

A self-signed certificate is generated.

For obtaining CA-signed certificates, you need to additionally generate and submit a CSR file to a CA. For more information, see 10.3.2 Generating a CA-Signed Certificate.

10.3.2 Generating a CA-Signed Certificate

Caution

NNMi 11-50 or later version introduce a Public Key Cryptography Standards (PKCS) #12 repository to store certificates. The new PKCS #12 file-based certificate management technique is available for use as soon as you install a new instance of NNMi 11-50 or later version on a system. Environments upgraded from an older version of NNMi continue to use a JKS repository to store certificates.

In upgraded environments, you can migrate to the PKCS #12 repository by using the steps in 10.2 Configuring an Upgraded NNMi Environment to Use the New Keystore.

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

- 1. Generate a self-signed certificate. For details, see 10.3.1 Generating a Self-Signed Certificate.
- 2. Run the following command to create a CSR (Certificate Signing Request) file:
 - Windows:

```
%NnmInstallDir%bin\nnmkeytool.ovpl -keystore nnm-key.p12 -certreq -storetype PKC
S12 -storepass nnmkeypass -alias <alias_name> -file CERTREQFILE
```

• Linux:

```
$NnmInstallDir/bin/nnmkeytool.ovpl -keystore nnm-key.p12 -certreq -storetype PKC
$12 -storepass nnmkeypass -alias <alias_name> -file CERTREQFILE
```

Note

- In the command above, <alias_name> corresponds to the alias you had provided at the time of generating the certificate.
- To print the contents of CERTREQFILE, run the following command.
 - Windows:
```
%NnmInstallDir%bin\nnmkeytool.ovpl -printcertreq -file CERTREQFILE -
storetype PKCS12
```

• Linux:

```
$NnmInstallDir/bin/nnmkeytool.ovpl -printcertreq -file CERTREQFILE -
storetype PKCS12
```

3. Send the CSR to your CA signing authority which signs and returns the certificate files. For information on different types of CA certificates, see (1) Types of CA-Signed Certificates.

The CA signing authority returns one of the following:

- A single signed server certificate file (referred to as myserver.crt file in this section). The single file contains the server certificate (the NNMi certificate that is CA-signed), one or more intermediate CA certificates, and the root CA certificate. All the certificates in this single file form a certificate chain.
- A set of two files that includes a signed server certificate file (referred to as myserver.crt file in this section) and a separate file containing the CA certificates (referred to as the myca.crt file). The myserver.crt file contains either a single server certificate or a certificate chain, but NOT the root CA certificate, which remains in the myca.crt file.



If your CA returns the certificates in other forms, contact the CA provider for more information about how to obtain the separate certificate chain and root CA certificate. NNMi supports PEM (Privacy Enhanced Mail) format certificates only. Please get PEM format certificates.

4. Prepare the certificate files.

The certificate chain must be imported to the keystore file and the root CA certificate must be imported to the truststore file.

• If you received a single file from step 3

Copy the root CA certificates from that file into a separate myca.crt file.

• If you received a set of two files from step 3

Add the myca.crt (the root CA certificate) file content to the end of the myserver.crt file and also remove any extra intermediate certificates from the myca.crt file, if it has any. This should result in one file, myserver.crt, containing the full certificate chain and one file, myca.crt, containing the root CA certificate.

- 5. Copy the files containing these certificates to a location on the NNMi management server. For this example, copy the files to the following location:
 - Windows: %NnmDataDir%shared\nnm\certificates
 - Linux: \$NnmDataDir/shared/nnm/certificates
- 6. Change to the directory on the NNMi management server that contains the keystore and truststore files:
 - Windows: %NnmDataDir%shared\nnm\certificates
 - Linux: \$NnmDataDir/shared/nnm/certificates
- 7. Run the following command to import the certificate into the keystore file:
 - Windows:

```
%NnmInstallDir%bin\nnmkeytool.ovpl -importcert -trustcacerts -keystore nnm-key.p
12 -storetype PKCS12 -storepass nnmkeypass -alias <alias_name> -file <path_to_my
server.crt>
```

• Linux:

```
$NnmInstallDir/bin/nnmkeytool.ovpl -importcert -trustcacerts -keystore nnm-key.p
12 -storetype PKCS12 -storepass nnmkeypass -alias <alias_name> -file <path_to_my
server.crt>
```



In the above command,

- <path_to_myserver.crt> corresponds to the full path of the location where you have stored the CA-signed server certificate.
- <alias name> corresponds to the alias you had provided at the time of generating the certificate.
- 8. When prompted to trust the certificate, enter: y

Example output for importing a certificate into the keystore

The output from the command is of the form:

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

9. Run the following commands to import the root certificate into the truststore file:

• Windows:

```
%NnmInstallDir%bin\nnmkeytool.ovpl -import -alias <alias_name> -storetype PKCS12
-keystore nnm-trust.p12 -file <path to myca.crt> -storepass ovpass
```

Linux:

```
$NnmInstallDir/bin/nnmkeytool.ovpl -import -alias <alias_name> -storetype PKCS12
  -keystore nnm-trust.p12 -file <path_to_myca.crt> -storepass ovpass
```

Note

In the above command,

- <path_to_myca.crt> corresponds to the full path of the location where you have stored the root certificate.
- <alias name> corresponds to the alias you had provided at the time of generating the certificate.

10. Examine the contents of the truststore:

• Windows:

```
%NnmInstallDir%bin\nnmkeytool.ovpl -list -keystore nnm-trust.p12 -storetype PKCS 12 -storepass ovpass
```

• Linux:

^{10.} Working with Certificates for NNMi

```
$NnmInstallDir/bin/nnmkeytool.ovpl -list -keystore nnm-trust.p12 -storetype PKCS
12 -storepass ovpass
```

Example truststore output

The truststore output is of the form:

```
Keystore type: PKCS12
Keystore provider:BCFIPS
Your keystore contains 1 entry
nnmi_ldap, Nov 14, 2008, trustedCertEntry,
Certificate fingerprint (MD5):29:02:D7:D7:D7:29:02:29:02:29:02:29:02:29:02
```

Q Тір

The truststore can include multiple certificates.

(1) Types of CA-Signed Certificates



Note

If your CA returns the certificates in other forms, contact the CA provider for instructions about obtaining the certificate chain and the Root CA Certificate.

The Certificate Authority (CA) should provide you with one of the following:

• A signed server certificate file containing the server certificate (the NNMi certificate that is CA signed) and one or more CA certificates. This section refers to the signed server certificate as myserver.crt file.

A CA Certificate can be either of the following:

- Root CA Certificate: Identifies the authority that is trusted to sign certificates for servers and users.
- Intermediate CA Certificate: A certificate signed by either a root or intermediate CA that is itself an authority, rather than a server or user.



The list of certificates from the NNMi server certificate to the root CA certificate, including any intermediate CA certificates, is known as the certificate chain.

• A signed server certificate and a separate file containing one or more CA certificates. This section refers to the signed server certificate as myserver.crt file and the CA certificates as myca.crt file. The myserver.crt file should contain either a single server certificate or a certificate chain, but NOT the root CA certificate, which would be in the myca.crt file.

To configure NNMi with the new certificate, you must import the certificate chain into the nnm-key.p12 file and the root CA Certificate into the nnm-trust.p12 file. Use the myserver.crt file when importing the server certificate into the nnm-key.p12 file and the myca.crt file when importing the CA certificate into the nnm-trust.p12 file.

Note

If your CA returns the certificates in other forms, contact the CA provider for instructions about obtaining the separate certificate chain and root CA Certificate.

When provided with one file that contains a full certificate chain, copy the root CA certificate from that file into the myca.crt file. Use the myca.crt file to import into the nnm-trust.pl2 file so that NNMi trusts the CA that issued the certificate.

When provided two files, add the myca.crt file content to the end of the myserver.crt file, if the file does not include it, and also remove any extra intermediate certificates from the myca.crt file, if it has any. This should result in one file, myserver.crt, containing the full certificate chain and one file, myca.crt file, containing the root CA Certificate.

Note

When using a CA, only the root CA certificate is generally added to the nnm-trust.p12 file. Adding intermediate CA or server certificates to the nnm-trust.p12 file will cause those certificates to be explicitly trusted and not checked for additional information, such as revocation. Only add additional certificates to the nnm-trust.p12 file if your CA requires it.

The following examples show what the files received from a CA signing authority might look like:

Separate server and CA certificate files:

```
----BEGIN CERTIFICATE----
Sample/AVQQKExNQU0EgQ29ycG9yYXRpb24gTHRkMRAwDgYDVQQLEwdOZXR3b3JseGVSZXZvY2F0aW9uTGlzd
D9iYXN1P29iamVjdENsYXNzPWNSTERpc3RyaWJ1dGlw
```

```
TZImiZPyLGQBGRYDaW50MRIwEAYKCZImiZPyLGQBGRYCc2cxEzARBgNVBAMTCmNbpSo6o/76yShtT7Vrlfz+m
XjWyEHaIy/QLCpPebYhejHEg4dZgzWWT/lQt==
-----END CERTIFICATE-----
```

Combined server and CA certificates in one file:

```
----BEGIN CERTIFICATE----
Sample1/VQQKExNQU0EgQ29ycG9yYXRpb24gTHRkMRAwDgYDVQQLEwd0ZXR3b3JseGVSZXZvY2F0aW9uTGlzd
D9iYXN1P29iamVjdENsYXNzPWNSTERpc3RyaWJ1dGlw
TZImiZPyLGQBGRYDaW50MRIwEAYKCZImiZPyLGQBGRYCc2cxEzARBqNVBAMTCmNbpSo6o/76yShtT7Vrlfz+m
XjWyEHaIy/QLCpPebYhejHEg4dZgzWWT/lQt==
----END CERTIFICATE----
----BEGIN CERTIFICATE----
Sample2/Gh0dHA6Ly9jb3JwMWRjc2cyLnNnLmludC5wc2FnbG9iYWwuY29tL0N1cRaOCApwwggKYMB0GA1UdD
gQWBBSqaWZzCRcpvJWOFPZ/Be9b+QSPyDAfBgNVHSMC
Wp5Lz1ZJAOu1VHbPVdQnXnlBkx7V65niLoaT90Eqd6laliVlJHj7GBriJ90uvVGuBQagggEChoG9bGRhcDovL
y9DTj1jb3JwMWRjc2cyL==
----END CERTIFICATE----
```

10.3.3 Delete a Certificate from the NNMi Keystore

The NNMi keystore can hold only one certificate at a time. Before replacing or renewing a certificate on the NNMi management server, you must delete the existing certificate from the NNMi keystore.

To delete a certificate from the NNMi keystore:

- 1. Change to the directory on the NNMi management server that contains the nnm-key.p12 and nnm-trust.p12 files:
 - Windows: %NnmDataDir%shared\nnm\certificates
 - Linux: \$NnmDataDir/shared/nnm/certificates
- 2. Save a backup copy of the nnm-key.p12 file.
- 3. Examine the contents of the keystore, and then note down the alias of the existing certificate:
 - Windows:

```
%NnmInstallDir%bin\nnmkeytool.ovpl -list -keystore nnm-key.p12 -storetype PKCS12
-storepass nnmkeypass
```

• Linux:

```
$NnmInstallDir/bin/nnmkeytool.ovpl -list -keystore nnm-key.p12 -storetype PKCS12
-storepass nnmkeypass
```

- 4. Delete the existing certificate from keystore by running the following command:
 - Windows:

```
%NnmInstallDir%bin\nnmkeytool.ovpl -delete -keystore nnm-key.pl2 -storetype PKCS
12 -storepass nnmkeypass -alias <alias>
```

• Linux:

```
$NnmInstallDir/bin/nnmkeytool.ovpl -delete -keystore nnm-key.p12 -storetype PKCS
12 -storepass nnmkeypass -alias <alias>
```

Note

The alias, referred to as <alias> in this example, identifies the existing certificate.

5. Restart NNMi by running the following commands:

Note

Changes take effect only after restarting NNMi.

• Windows:

```
%NnmInstallDir%bin\ovstop -c
%NnmInstallDir%bin\ovstart -c
```

• Linux:

```
$NnmInstallDir/bin/ovstop -c
$NnmInstallDir/bin/ovstart -c
```

10.3.4 Replacing an Existing Certificate with a new Self-Signed or CA-Signed Certificate

A self-signed certificate is created and installed during NNMi installation. You would typically replace a certificate in any of the following scenarios:

- To use a new self-signed or CA-signed certificate instead of the default certificate.
- To renew an expired certificate.

To replace a certificate, do the following:

1. Generate a self-signed certificate. For details, see 10.3.1 Generating a Self-Signed Certificate.

Or, if you organization requires the certificate to be signed by a CA, generate a CSR (Certificate Signing Request) file and obtain a CA signed certificate. For details, see 10.3.2 Generating a CA-Signed Certificate.

2. Test HTTPS access to the NNMi console using the following syntax:

https://<fully_qualified_domain_name>:<port_number>/nnm/.

If you have used CA-signed certificate and if the browser trusts the CA, it will trust the HTTPS connection to the NNMi console.

If you have used self-signed certificate, browser displays a warning message about the untrusted HTTPS connection to the NNMi Console.

10.3.5 Working with Certificates in Application Failover Environments

Figure 10-1: Using certificates with application failover



Caution

NNMi 11-50 or later version introduce a Public Key Cryptography Standards (PKCS) #12 repository to store certificates. The new PKCS #12 file-based certificate management technique is available for use as soon as you install a new instance of NNMi 11-50 or later version on a system. Environments upgraded from an older version of NNMi continue to use a JKS repository to store certificates.

In upgraded environments, you can migrate to the PKCS #12 repository by using the steps in 10.2 Configuring an Upgraded NNMi Environment to Use the New Keystore.

When configuring the application failover feature, you must merge the content of the truststore file for both nodes into one nnm-trust.pl2 file.

Complete the following steps to configure the application failover feature to use self-signed or CA-signed certificates.

Caution

If you are using self-signed certificates with NNMi along with the application failover feature, and do not complete the following steps, NNMi processes will not start correctly on the standby NNMi management server (Server Y in this example).

- 1. Change to the following directory on Server Y:
 - Windows: %NnmDataDir%shared\nnm\certificates
 - Linux: \$NnmDataDir/shared/nnm/certificates
- 2. Copy the nnm-trust.p12 file from Server Y to some temporary location on Server X.

The remaining steps refer to these file locations as <truststore>.

3. Run the following command on Server X to merge Server Y's truststore into Server X's nnm-trust.p12 file:

nnmcertmerge.ovpl -truststore <truststore>

- 4. Copy the merged nnm-trust.p12 file from Server X to Server Y, so that both nodes have the merged files. The location of this file is as follows:
 - Windows: %NnmDataDir%shared\nnm\certificates
 - Linux: \$NnmDataDir/shared/nnm/certificates
- 5. Run the following command on both Server X and Server Y.

Verify that the displayed results from both servers, including the fully-qualified-domain names, match. If they do not match, do not continue; instead, redo beginning with step 1. Windows:

```
10. Working with Certificates for NNMi
```

```
%NnmInstallDir%bin\nnmkeytool.ovpl -list -keystore
%NnmDataDir%shared\nnm\certificates\nnm-trust.p12 -storetype PKCS12 -storepass ovp
ass
```

Linux:

```
$NnmInstallDir/bin/nnmkeytool.ovpl -list -keystore
$NnmDataDir/shared/nnm/certificates/nnm-trust.p12 -storetype PKCS12 -storepass ovp
ass
```

6. Continue configuring the application failover feature at 18. Configuring NNMi for Application Failover.

10.3.6 Working with Certificates in High-Availability Environments

This section describes how to configure NNMi to use Self-Signed or Certificate Authority Certificates in an HA environment.

Figure 10-2: Using certificates with HA





Caution

NNMi 11-50 or later version introduce a Public Key Cryptography Standards (PKCS) #12 repository to store certificates. The new PKCS #12 file-based certificate management technique is available for use as soon as you install a new instance of NNMi 11-50 or later version on a system. If you have upgraded from an older version of NNMi, you must migrate to the PKCS #12 repository manually.

In upgraded environments, you can migrate to the PKCS #12 repository by using the steps in 10.2 Configuring an Upgraded NNMi Environment to Use the New Keystore.

(1) Configuring High-Availability Using Default Certificates

The process for configuring NNMi for HA correctly shares the default self-signed certificate among the primary and secondary cluster nodes. You do not need to take any extra steps to use the default certificate with NNMi running under HA.

(2) Configuring High-Availability Using New Certificates

This section creates a new self-signed or CA certificate, referred to as newcert. Complete the following steps to configure HA with this new CA or self-signed certificate.



Important

When making file changes under High Availability (HA), you must make the changes on both nodes in the cluster. If the change requires you to stop and restart the NNMi management server, you must put the nodes in maintenance mode before running the ovstop and ovstart commands. See 19.6.1 Placing NNMi in maintenance mode for more information.

) Тір

You can complete this procedure before or after configuring NNMi for HA, as described in 19.5 Shared NNMi Data.

- 1. Change to the following directory on NNMi_HA1 before completing step 2:
 - Windows: %NnmDataDir%shared\nnm\certificates
 - Linux: \$NnmDataDir/shared/nnm/certificates
- 2. On NNMi_HA1, run the following commands to import newcert into the nnm-key.p12 file:

Windows:

```
%NnmInstallDir%bin\nnmkeytool.ovpl -import -alias <newcert_Alias> -storetype PKCS1
2 -keystore nnm-key.p12 -file newcert -storepass nnmkeypass
```

Linux:

```
$NnmInstallDir/bin/nnmkeytool.ovpl -import -alias <newcert_Alias> -storetype PKCS1
2 -keystore nnm-key.p12 -file newcert -storepass nnmkeypass
```

10.3.7 Working with Certificates in Global Network Management Environments

Caution

NNMi 11-50 or later version introduce a Public Key Cryptography Standards (PKCS) #12 repository to store certificates. The new PKCS #12 file-based certificate management technique is available for use as soon as you install a new instance of NNMi 11-50 or later version on a system. Environments upgraded from an older version of NNMi continue to use a JKS repository to store certificates.

In upgraded environments, you can migrate to the PKCS #12 repository by using the steps in 10.2 Configuring an Upgraded NNMi Environment to Use the New Keystore.

During NNMi installation, the installation script creates a self-signed certificate for the NNMi management server. This certificate contains an alias that includes the fully-qualified domain name of the node. The installation script adds this self-signed certificate to the NNMi management server's nnm-key.pl2 and nnm-trust.pl2 files.

Complete the following steps to configure the global network management feature to use self-signed/CA-signed certificates based on the following diagram.

Before you begin, make sure that the required certificates are created on the regional manager systems. For details, see 10.3.4 Replacing an Existing Certificate with a new Self-Signed or CA-Signed Certificate.

^{10.} Working with Certificates for NNMi

Note

Ë

If you are using a mix of newly installed NNMi 11-50 or later version instances and NNMi management servers upgraded to 11-50 or later version from an older version, follow the guideline in NNMi management servers upgraded to the version 11-50.

Figure 10-3: Global network management



- 1. Change to the following directory on regional1 and regional2:
 - Windows: %NnmDataDir%shared\nnm\certificates
 - Linux: \$NnmDataDir/shared/nnm/certificates
- 2. Copy the nnm-trust.p12 files from the above locations on regional1 and regional2 to some temporary location on global1.
- 3. Run the following commands on global1 to merge the regional1 and regional2 certificates into global1's nnm-trust.p12 file:

```
nnmcertmerge.ovpl -truststore <regional1_nnm-trust.p12_location>
nnmcertmerge.ovpl -truststore <regional2_nnm-trust.p12_location>
```

4. Run the following commands to restart NNMi on global1:

```
ovstop
ovstart
```

Important

When making file changes under High Availability (HA), you need to make the changes on both nodes in the cluster. For NNMi using HA configurations, if the change requires you to stop and restart the NNMi management server, you must put the nodes in maintenance mode before running the ovstop and ovstart commands. See 19.6.1 Placing NNMi in maintenance mode for more information.

10.3.8 Configuring an SSL connection to the Directory service

Caution

NNMi 11-50 or later version introduce a Public Key Cryptography Standards (PKCS) #12 repository to store certificates. The new PKCS #12 file-based certificate management technique is available for use as soon as you install a new instance of NNMi 11-50 or later version on a system. Environments upgraded from an older version of NNMi continue to use a JKS repository to store certificates.

In upgraded environments, you can migrate to the PKCS #12 repository by using the steps in 10.2 Configuring an Upgraded NNMi Environment to Use the New Keystore.

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

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

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

1. Obtain your company's truststore certificate from the directory server.

The directory service administrator should be able to give you a copy of this text file.

Note that the owner CN of the Truststore certificate must match the host name of the directory server.

- 2. Change to the directory that contains the NNMi truststore:
 - Windows: %NnmDataDir%shared\nnm\certificates
 - Linux: \$NnmDataDir/shared/nnm/certificates

Run all commands in this procedure from the certificates directory.

3. Import your company's truststore certificate into the NNMi truststore.

Note

Import the root CA certificate of the LDAP directory server (without intermediate certificates) into the NNMi truststore.

If you need to import root CA certificates for multiple LDAP directory servers, when you import the second and subsequent certificates, replace "nnmi_ldap" in the procedure with a name of your choice (example: nnmi_ldap2).

a. Run the following command:

Windows:

```
%NnmInstallDir%bin\nnmkeytool.ovpl -import -alias nnmi_ldap -storetype PKCS12 -k
eystore nnm-trust.p12 -storepass ovpass -file <Directory Server Certificate.txt>
```

Linux:

```
$NnmInstallDir/bin/nnmkeytool.ovpl -import -alias nnmi_ldap -storetype PKCS12 -k
eystore nnm-trust.p12 -storepass ovpass -file <Directory_Server_Certificate.txt>
```

Where <Directory_Server_Certificate.txt> is your company's truststore certificate.

^{10.} Working with Certificates for NNMi

b. When prompted to trust the certificate, enter: y

Example output for importing a certificate into the truststore

The output from this command is of the form:

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

4. Examine the contents of the truststore:

Windows:

```
%NnmInstallDir%bin\nnmkeytool.ovpl -list -storetype PKCS12 -keystore nnm-trust.p12 -storepass ovpass
```

Linux:

```
$NnmInstallDir/bin/nnmkeytool.ovpl -list -storetype PKCS12 -keystore nnm-trust.p12
-storepass ovpass
```

Example truststore output

The truststore output is of the form:

```
Keystore type:PKCS12
Keystore provider:SunJSSE
Your keystore contains 1 entry
nnmi_ldap, Nov 14, 2008, trustedCertEntry,
Certificate fingerprint (MD5):29:02:D7:D7:D7:D7:29:02:29:02:29:02:29:02:29:02
```



The truststore can include multiple certificates.

5. Run the following commands to restart NNMi:

ovstop ovstart

Important

When making file changes under High Availability (HA), you must make the changes on both nodes in the cluster. If the change requires you to stop and restart the NNMi management server, you must put the nodes in maintenance mode before running the ovstop and ovstart commands. See 19.6.1 Placing NNMi in maintenance mode for more information.

10.4 Using Certificates with the JKS Repository

Prior to the version 11-50, NNMi used to provide a Java KeyStore (JKS) repository to store certificates. NNMi 11-50 or later version introduce a Public Key Cryptography Standards (PKCS) #12 repository to store certificates. The new PKCS #12 file-based certificate management technique is available for use as soon as you install a new instance of NNMi 11-50 or later version on a system.

However, when you upgrade an older version of NNMi to the version 11-50 or later version, the PKCS #12 file-based certificate management does not immediately come into effect and NNMi continues to use the JKS repository for certificate management.

If you like, you can continue with the older JKS repository of certificates. This section provides you with instructions to use certificates when you want to continue to use the JKS repository of certificates.

Do not use the information in this section in the following two scenarios:

- You have installed a new instance of NNMi
- You have upgraded NNMi from an older version, but you have performed the steps in the 10.2 Configuring an Upgraded NNMi Environment to Use the New Keystore

Concept	Description
Keystore and Truststore	<pre>Truststore : NNMi truststore is the nnm.truststore file in which you store public keys from sources that you want NNMi to trust. Keystore : NNMi keystore is the nnm.keystore file in which you import NNMi server's private key. The nnm.truststore and nnm.keystore files are located at: Windows: %NNM_DATA%\shared\nnm\certificates\ Linux: \$NNM_DATA/shared/nnm/certificates/</pre>
Default NNMi certificates	NNMi is installed with a self-signed certificate generated using default properties. You can replace the default certificate with another self-signed or CA-signed certificate.
Tools	Certificates are generated and managed using Java's Keytool utility. Additionally, NNMi provides the nnmmergecert.ovpl utility to merge certificates to establish trust within NNMi systems. This program is used in HA, Failover, and Global Network Management setups.
Supported encryption algorithms	NNMi accepts certificates generated using RSA algorithm. DSA algorithm is not supported.
Self-Signed Certificate	A Self-Signed certificate is typically used for establishing secure communication between your server and a known group of clients. NNMi installs with a self-signed certificate generated using default properties. Note: NNMi instances configured to use a self-signed certificate will display a warning message when users try to access NNMi web console in a web browser.
CA-Signed Certificate	Signed server certificate that you receive in response to the Certificate Signing Request will contain the NNMi certificate that is CA signed and one or more CA certificates (if there is more than one CA certificate, this is also known as the certificate chain). Note: These certificates might be in a single file or in a two separate files.
Root CA Certificate	Identifies the certificate authority that is trusted to sign certificates for servers and users.
Intermediate CA Certificate	A certificate signed by either a root or intermediate CA that is itself an authority, rather than a server or user. Note: The list of certificates from the NNMi server certificate to the root CA certificate, including any intermediate CA certificates, is known as the certificate chain.

Table 10-2: Certificate Terminology

10.4.1 Replacing an Existing Certificate with a new Self-Signed or CA-Signed Certificate

A self-signed certificate is created and installed during NNMi installation. You would typically replace a certificate in any of the following scenarios:

- To use a new self-signed or CA-signed certificate instead of the default certificate.
- To renew an expired certificate.

To replace a certificate, do the following:

- 1. Generate a self-signed certificate. For details, see 10.4.2 Generating a Self-Signed Certificate.
- 2. If you organization requires the certificate to be signed by a CA, generate a CSR (Certificate Signing Request) file and obtain a CA signed certificate. For details, see 10.4.3 Generating a CA-Signed Certificate.
- 3. Open the following file and update the com.hp.ov.nms.ssl.KEY_ALIAS variable to the value you used for <alias> while generating a certificate.
 - Windows: %NNM CONF%\nnm\props\nms-local.properties
 - Linux: \$NNM CONF/nnm/props/nms-local.properties

4. Restart the NNMi Management Server.

```
ovstop
ovstart
```

Note

When making file changes under High Availability (HA), you need to make the changes on both nodes in the cluster. For NNMi using HA configurations, if the change requires you to stop and restart the NNMi management server, you must put the nodes in maintenance mode before running the ovstop and ovstart commands.

5. Test HTTPS access to the NNMi console using the following syntax:

https://<fully_qualified_domain_name>:<port_number>/nnm/.

If you have used CA-signed certificate and if the browser trusts the CA, it will trust the HTTPS connection to the NNMi console.

If you have used self-signed certificate, browser displays a warning message about the untrusted HTTPS connection to the NNMi Console.

10.4.2 Generating a Self-Signed Certificate

To generate a self-signed certificate, follow these steps:

- 1. Change to the directory on the NNMi management server that contains the nnm.keystore and nnm.truststore files:
 - Windows: %NnmDataDir%shared\nnm\certificates
 - Linux: \$NnmDataDir/shared/nnm/certificates
- 2. Save a backup copy of the nnm. keystore file.

10. Working with Certificates for NNMi

Important

- If you are replacing an existing NNMi certificate, do not remove the existing certificate until you complete these steps. NNMi must start up at least once with both the old and new certificate installed so that it can transfer encrypted information to the new certificate.
- Make sure the alias points to the new certificate as described in the next step to ensure NNMi presents the new certificate on the NNMi management server to the client servers.
- 3. Generate a private key from your system. Use the keytool command to generate this private key:

a. Run the following command exactly as shown:

• Windows:

```
%jdkdir%\bin\keytool.exe -genkeypair -validity 36500 -keyalg rsa -keystore nn
m.keystore -storepass nnmkeypass -alias <alias name>
```

• Linux:

```
$jdkdir/bin/keytool -genkeypair -validity 36500 -keyalg rsa -keystore nnm.key
store -storepass nnmkeypass -alias <alias name>
```

Note

The alias, referred to as <alias_name> in this example, identifies this newly-created key. Although the alias can be any string, we recommends you use the fully-qualified domain name (FQDN) followed by a suffix to help you easily identify the right version.

b. Enter the requested information.



When prompted for your first and last name, enter the FQDN of your system.

A self-signed certificate is generated.

For obtaining CA-signed certificates, you need to additionally generate and submit a CSR file to a CA. For more information, see 10.4.3 Generating a CA-Signed Certificate.

10.4.3 Generating a CA-Signed Certificate

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

- 1. Generate a self-signed certificate. For details, see 10.4.2 Generating a Self-Signed Certificate.
- 2. Run the following command to create a CSR (Certificate Signing Request) file:
 - Windows:

```
%jdkdir%\bin\keytool.exe -keystore nnm.keystore -certreq -storepass nnmkeypass -
alias <alias name> -file CERTREQFILE
```

• Linux:

```
$jdkdir/bin/keytool -keystore nnm.keystore -certreq -storepass nnmkeypass -alias
  <alias_name> -file CERTREQFILE
```

Note

For more information about the keytool command, search for *Key and Certificate Management Tool* at Oracle homepage.

- 3. Send the CSR to your CA signing authority which signs and returns the certificate files. For information on different types of CA certificates, see (1) Types of CA-Signed Certificates.
- 4. Copy the files containing these certificates to a location on the NNMi management server. For this example, copy the files to the following location:
 - Windows: %NnmDataDir%shared\nnm\certificates
 - Linux: \$NnmDataDir/shared/nnm/certificates
- 5. Change to the directory on the NNMi management server that contains the nnm.keystore and nnm.truststore files:
 - Windows: %NnmDataDir%shared\nnm\certificates
 - Linux: \$NnmDataDir/shared/nnm/certificates
- 6. Run the following command to import the certificate into the nnm.keystore file:
 - Windows:

```
%jdkdir%\bin\keytool.exe -importcert -trustcacerts -keystore nnm.keystore -store
pass nnmkeypass -alias <alias_name> -file <myserver.crt>
```

• Linux:

```
$jdkdir/bin/keytool -importcert -trustcacerts -keystore nnm.keystore -storepass
nnmkeypass -alias <alias_name> -file <myserver.crt>
```

Note

- In the above command,
 - <myserver.crt> corresponds to the full path of the location where you have stored the signed server certificate.
 - <alias_name> corresponds to the alias you had provided at the time of generating the certificate.
- If you use the -storepass option and provide the password, the keystore program does not prompt you for the keystore password. If you do not use the -storepass option, enter nnmkeypass when prompted for the keystore password.
- 7. When prompted to trust the certificate, enter: y

Example output for importing a certificate into the keystore

The output from the command is of the form:

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

- 8. Run the following commands to import the certificate into the nnm.truststore file:
 - Windows:

```
%jdkdir%\bin\keytool.exe -import -alias <alias_name> -keystore nnm.truststore -f
ile <myca.crt>
```

• Linux:

```
$jdkdir/bin/keytool -import -alias <alias_name> -keystore nnm.truststore -file <
myca.crt>
```

Note

- In the above command,
 - <myserver.crt> corresponds to the full path of the location where you have stored the signed server certificate.
 - <alias_name> corresponds to the alias you had provided at the time of generating the certificate.
- If you use the -storepass option and provide the password, the keystore program does not prompt you for the keystore password. If you do not use the -storepass option, enter nnmkeypass when prompted for the keystore password.
- 9. When prompted for the truststore password, enter: ovpass.

10. Examine the contents of the truststore:

• Windows:

%jdkdir%\bin\keytool.exe -list -keystore nnm.truststore

• Linux:

\$jdkdir/bin/keytool -list -keystore nnm.truststore

When prompted for the truststore password, enter: ovpass

Example truststore output

The truststore output is of the form:

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

О Тір

The truststore can include multiple certificates.

(1) Types of CA-Signed Certificates



Note

If your CA returns the certificates in other forms, contact the CA provider for instructions about obtaining the certificate chain and the Root CA Certificate. NNMi supports PEM (Privacy Enhanced Mail) format certificates only. Please get PEM format certificates.

The Certificate Authority (CA) should provide you with one of the following:

• A signed server certificate file containing the server certificate (the NNMi certificate that is CA signed) and one or more CA certificates. This section refers to the signed server certificate as myserver.crt.

A CA Certificate can be either of the following:

- Root CA Certificate Identifies the authority that is trusted to sign certificates for servers and users.
- Intermediate CA Certificate A certificate signed by either a root or intermediate CA that is itself an authority, rather than a server or user.



The list of certificates from the NNMi server certificate to the root CA certificate, including any intermediate CA certificates, is known as the certificate chain.

• A signed server certificate and a separate file containing one or more CA certificates. This section refers to the signed server certificate as myserver.crt and the CA certificates as myca.crt. The myserver.crt file

10. Working with Certificates for NNMi

should contain either a single server certificate or a certificate chain, but NOT the root CA certificate, which would be in the myca.crt file.

To configure NNMi with the new certificate, you must import the certificate chain into the nnm.keystore and the root CA Certificate into the nnm.truststore. Use the myserver.crt file when importing the server certificate into the nnm.keystore file and the myca.crt file when importing the CA certificate into the nnm.truststore file.



The list of certificates from the NNMi server certificate to the root CA certificate, including any intermediate CA certificates, is known as the certificate chain.

When provided with one file that contains a full certificate chain, copy the root CA certificate from that file into the myca.crt file. Use the myca.crt file to import into the nnm.truststore so that NNMi trusts the CA that issued the certificate.

When provided two files, add the myca.crt file content to the end of the myserver.crt, if the file does not include it, and also remove any extra intermediate certificates from the myca.crt, if it has any. This should result in one file, myserver.crt, containing the full certificate chain and one file, myca.crt, containing the root CA Certificate.

Note

When using a CA, only the root CA certificate is generally added to the nnm.truststore. Adding intermediate CA or server certificates to the nnm.truststore will cause those certificates to be explicitly trusted and not checked for additional information, such as revocation. Only add additional certificates to the nnm.truststore if your CA requires it.

The following examples show what the files received from a CA signing authority might look like:

Separate server and CA certificate files:

```
-----BEGIN CERTIFICATE-----
Sample/AVQQKExNQU0EgQ29ycG9yYXRpb24gTHRkMRAwDgYDVQQLEwd0ZXR3b3JseGVSZXZvY2F0aW9uTGlzd
D9iYXN1P29iamVjdENsYXNzPWNSTERpc3RyaWJ1dGlw
.....
TZImiZPyLGQBGRYDaW50MRIwEAYKCZImiZPyLGQBGRYCc2cxEzARBgNVBAMTCmNbpSo6o/76yShtT7Vrlfz+m
XjWyEHaIy/QLCpPebYhejHEg4dZgzWWT/lQt==
-----END CERTIFICATE-----
```

Combined server and CA certificates in one file:

```
-----BEGIN CERTIFICATE-----
Sample1/VQQKExNQU0EgQ29ycG9yYXRpb24gTHRkMRAwDgYDVQQLEwdOZXR3b3JseGVSZXZvY2F0aW9uTGlzd
D9iYXN1P29iamVjdENsYXNzPWNSTERpc3RyaWJ1dGlw
.....
TZImiZPyLGQBGRYDaW50MRIwEAYKCZImiZPyLGQBGRYCc2cxEzARBgNVBAMTCmNbpSo6o/76yShtT7Vrlfz+m
XjWyEHaIy/QLCpPebYhejHEg4dZgzWWT/lQt==
-----END CERTIFICATE-----
Sample2/Gh0dHA6Ly9jb3JwMWRjc2cyLnNnLmludC5wc2FnbG9iYWwuY29tL0NlcRaOCApwwggKYMB0GA1UdD
gQWBBSqaWZzCRcpvJWOFPZ/Be9b+QSPyDAfBgNVHSMC
```

```
Wp5Lz1ZJAOu1VHbPVdQnXnlBkx7V65niLoaT90Eqd6laliVlJHj7GBriJ90uvVGuBQagggEChoG9bGRhcDovL
y9DTj1jb3JwMWRjc2cyL==
----END CERTIFICATE----
```

10.4.4 Configuring application failover to use self-signed certificates

Figure 10-4: Using self-signed certificates with application failover



When configuring the application failover feature, you must merge the contents of both nodes' nnm.keystore files and nnm.truststore files into a single nnm.keystore file and a single nnm.truststore file, respectively. Complete the following steps to configure the application failover feature to use self-signed certificates based on the above diagram.



Caution

If you are using self-signed certificates with NNMi together with the application failover feature and you do not perform the following steps, NNMi processes will not start correctly on the standby NNMi management server (Server Y in this example).

- 1. Change to the following directory on Server Y:
 - Windows: %NNM DATA%\shared\nnm\certificates
 - Linux: \$NNM DATA/shared/nnm/certificates
- 2. Copy the nnm.keystore and nnm.truststore files from Server Y to some temporary location on Server X. The remaining steps refer to these file locations as <keystore> and <truststore>.
- 3. Execute the following command on Server X to merge Server Y's certificates into Server X's nnm. keystore and nnm.truststore files:

```
nnmcertmerge.ovpl -keystore <keystore> -truststore <truststore>
```

4. Copy the merged nnm. keystore and nnm. truststore files from Server X to Server Y, so that both nodes have the merged files.

The location of these files is as follows:

- Windows: %NNM DATA%\shared\nnm\certificates
- Linux: \$NNM DATA/shared/nnm/certificates
- 5. Execute the command shown below on both Server X and Server Y.

Verify that the displayed results from both servers, including the fully-qualified-domain names, match. If they do not match, do not continue; instead, redo beginning with step 1. Windows:

^{10.} Working with Certificates for NNMi

```
%jdkdir%\bin\keytool.exe -list -keystore
%NnmDataDir%shared\nnm\certificates\nnm.keystore -storepass nnmkeypass
```

Linux:

```
$jdkdir/bin/keytool -list -keystore
$NnmDataDir/shared/nnm/certificates/nnm.keystore -storepass nnmkeypass
```

6. Execute the command shown below on both Server X and Server Y.

Verify that the displayed results from both servers, including the fully-qualified-domain names, match. If they do not match, do not continue; instead, redo beginning with step 1.

Windows:

```
%jdkdir%\bin\keytool.exe -list -keystore
%NnmDataDir%shared\nnm\certificates\nnm.truststore -storepass ovpass
```

Linux:

```
$jdkdir/bin/keytool -list -keystore
$NnmDataDir/shared/nnm/certificates/nnm.truststore -storepass ovpass
```

7. Continue configuring the application failover feature from 18. Configuring NNMi for Application Failover.



Although you manually completed step 4, after you start the application failover feature, NNMi automatically replicates the merged keystore and truststore information from Server X to Server Y.

10.4.5 Working with Certificates in High-Availability Environments

Figure 10-5: Using certificates with HA



This section explains how to configure a high availability cluster that uses self-signed or Certificate Authority certificates based on the above diagram.

(1) Configuring High-Availability Using Default Certificates

The process for configuring NNMi for HA correctly shares the self-signed certificate among the primary and secondary cluster nodes. You do not need to take any extra steps to use the default certificate with NNMi running under HA.

(2) Configuring an HA cluster to use a new certificate

Suppose you create a new self-signed or CA certificate, referred to as newcert. Complete the following steps to configure HA to use this new CA or self-signed certificate.

You can complete this procedure before or after configuring NNMi for HA, as described in 19.4 Configuring HA.

Important

When making file changes under High Availability (HA), you must make the changes on both nodes in the cluster. If the change requires you to stop and restart the NNMi management server, you must put the nodes in maintenance mode before running the ovstop and ovstart commands. See 19.6.1 Placing NNMi in maintenance mode for more information.

1. Change to the following directory on NNMi_HA1 before completing step 2:

- Windows: %NNM_DATA%\shared\nnm\certificates
- Linux: \$NNM_DATA/shared/nnm/certificates
- 2. On NNMi_HA1, execute the following command to import newcert into the nnm.keystore file: Windows:

```
%jdkdir%\bin\keytool.exe -import -alias <newcert_Alias> -keystore nnm.keystore -fi
le newcert
```

Linux:

```
$jdkdir/bin/keytool -import -alias <newcert_Alias> -keystore nnm.keystore -file ne
wcert
```

- 3. Edit the following file on both the active cluster node (NNMi_HA1) and the standby node (NNMi_HA2):
 - Windows: %NNM_DATA%\conf\nnm\props\nms-local.properties
 - Linux: \$NNM_DATA/conf/nnm/props/nms-local.properties
- 4. Change the following line in the nms-local.properties file on both NNMi_HA1 and NNMi_HA2:

com.hp.ov.nms.ssl.KEY_ALIAS = <newcert_Alias>

5. Save your changes.

10.4.6 Working with Certificates in Global Network Management Environments

During NNMi installation, the installation script creates a self-signed certificate for the NNMi management server. This certificate contains an alias that includes the fully-qualified domain name of the node. The installation script adds this self-signed certificate to the NNMi management server's nnm.keystore and nnm.truststore files.

Suppose you want your global network management configuration to model the following figure.

Figure 10-6: Global network management



Complete the steps below to configure the global network management feature to use self-signed certificates based on the above figure.

- 1. Change to the following directory on regional1 and regional2:
 - Windows: %NNM_DATA%\shared\nnm\certificates
 - Linux: \$NNM_DATA/shared/nnm/certificates
- 2. Copy the nnm.truststore files from the above locations on regional1 and regional2 to some temporary location on global1.
- 3. Execute the following commands on globall to merge the regional1 and regional2 certificates into global1's nnm.truststore file:

```
nnmcertmerge.ovpl -truststore <regional1_nnm.truststore_location>
nnmcertmerge.ovpl -truststore <regional2_nnm.truststore_location>
```

4. Execute the following command sequence on global1:

```
ovstop
ovstart
```

10.4.7 Configuring an SSL connection to the directory service

By default, when directory service communications are enabled, NNMi uses the LDAP protocol for retrieving data from a directory service. If your directory service requires an SSL connection, you must enable the SSL protocol to encrypt the data that flows between NNMi and the directory service. To enable the SSL protocol, see 12.2.3 Task 3: Configure user access from the directory service.

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

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

1. Obtain your company's truststore certificate from the directory server.

The directory service administrator can give you a copy of this text file.

- 2. Change to the directory that contains the NNMi truststore:
 - Windows: %NNM DATA%\shared\nnm\certificates
 - Linux: \$NNM_DATA/shared/nnm/certificates

Execute all commands in this procedure from the certificates directory.

- 3. Import your company's truststore certificate into the NNMi truststore.
 - a. Execute the following command:

Windows:

```
%jdkdir%\bin\keytool.exe -import -alias nnmi_ldap -keystore nnm.truststore -file
<Directory Server Certificate.txt>
```

Linux:

```
$jdkdir/bin/keytool -import -alias nnmi_ldap -keystore nnm.truststore -file <Dir
ectory_Server_Certificate.txt>
```

<Directory Server Certificate.txt> is your company's truststore certificate.

b. When prompted for the truststore password, enter ovpass.

c. When prompted to trust the certificate, enter y.

Example output for importing a certificate into the truststore

The output format of this command is as follows:

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

4. Check the contents of the truststore:

Windows:

%jdkdir%\bin\keytool.exe -list -keystore nnm.truststore

Linux:

\$jdkdir/bin/keytool -list -keystore nnm.truststore

When prompted for the truststore password, enter ovpass.

Example truststore output

The truststore output format is as follows:

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

🕽 Тір

The truststore can include multiple certificates.

5. Execute the following commands to restart NNMi:

```
ovstop
ovstart
```

For more information about the keytool command, search for *Key and Certificate Management Tool* at Oracle homepage.

^{10.} Working with Certificates for NNMi

11

Configuring the Telnet and SSH Protocols for Use by NNMi

Choosing the Actions > Node Access > Telnet... (from client) menu item invokes the telnet command to the selected node (from the Web browser in which the NNMi console is currently running). Choosing the Actions > Node Access > Secure Shell... (from client) menu item invokes the secure shell (SSH) command to the selected node. By default, neither Internet Explorer, Microsoft Edge nor Mozilla Firefox defines the telnet command or the SSH command, so using either of these menu items produces an error message. You can configure the telnet, SSH, or both protocols for each NNMi user (on a per-system basis), and you can change the NNMi console menu items. This chapter explains the configuration of the telnet and SSH protocols used in NNMi.

11.1 Disable the telnet or SSH menu item

If the NNMi users in your deployment environment do not require telnet or SSH connection from the NNMi console, you can disable the respective menu item to remove it from the NNMi console.

Disablement of a menu item on the NNMi console is applicable to all users who sign in to the NNMi console on this NNMi management server. To disable the **Telnet** or **Secure Shell** menu item, follow these steps:

- 1. In the Configuration workspace, expand User Interface, and then select Menu Items.
- 2. In the Menu Items view, double-click the Telnet... (from client) row or the Secure Shell...(from client) row.
- 3. On the **Menu Item** form, clear the **Enabled** check box, and then set the **Author** field to an appropriate value. Changing the author value ensures that this menu item remains disabled when you upgrade NNMi.
- 4. Save and close the form.

For details, see Control the NNMi Console Menus in NNMi Help.

^{11.} Configuring the Telnet and SSH Protocols for Use by NNMi

11.2 Configure a telnet or SSH client for the browser on Windows

Configure the telnet command provided by the operating system for an NNMi user's Web browser. This procedure must be performed for each computer and Web browser from which an NNMi user needs to run the Actions > Node Access > Telnet... (from client) menu item.

Configure a third-party SSH command for an NNMi user's Web browser. This procedure must be performed for each computer and Web browser from which an NNMi user needs to run the Actions > Node Access > Secure Shell... (from client) menu item.

To complete any of the procedures in this section, you must have administrative privileges on the computer. The specific steps depend on the version (32-bit or 64-bit) of the browser and the operating system.

To determine the version of Internet Explorer, click **Help**, and then **About Internet Explorer**. If the version information does not include the text **64-bit Edition**, this Internet Explorer is 32-bit.

Only Microsoft Edge Chromium version is supported. To determine the version of Microsoft Edge, click the **Settings** and more button, click the **Help and Feedback**, and then click the **About Microsoft Edge**. If the version information includes the following sentence, this Microsoft Edge is Chromium edition."This browser is made possible by the Chromium open source project and other open source software."

Firefox is available only in a 32-bit version.

The following table identifies the procedure to use for each browser and operating system combination.

Web browser	Windows operating system architecture	Applicable procedures
Internet Explorer 32-bit	32-bit	 11.2.1 Windows operating system-provided telnet client 11.2.2 Third-party telnet client (standard Windows) 11.2.4 Third-party SSH client (standard Windows and Windows on Windows)
	64-bit Windows 7, Windows 10	 11.2.2 Third-party telnet client (standard Windows) 11.2.4 Third-party SSH client (standard Windows and Windows on Windows)
	64-bit other than Windows 7 or Windows 10	 11.2.3 Third-party telnet client (Windows on Windows) 11.2.4 Third-party SSH client (standard Windows and Windows on Windows)
Internet Explorer 64-bit	64-bit	 11.2.1 Windows operating system-provided telnet client 11.2.2 Third-party telnet client (standard Windows) 11.2.4 Third-party SSH client (standard Windows and Windows on Windows)
Microsoft Edge	64-bit	 11.2.1 Windows operating system-provided telnet client 11.2.2 Third-party telnet client (standard Windows) 11.2.4 Third-party SSH client (standard Windows and Windows on Windows)
Firefox	32-bit	 11.2.1 Windows operating system-provided telnet client 11.2.2 Third-party telnet client (standard Windows) 11.2.4 Third-party SSH client (standard Windows and Windows on Windows)
	64-bit Windows 7, Windows 10	• 11.2.2 Third-party telnet client (standard Windows)

Table 11-1: Matrix of telnet and SSH configuration procedures on Windows

11. Configuring the Telnet and SSH Protocols for Use by NNMi

Web browser	Windows operating system architecture	Applicable procedures
Firefox	64-bit Windows 7, Windows 10	• 11.2.4 Third-party SSH client (standard Windows and Windows on Windows)
	64-bit other than Windows 7 or Windows 10	 11.2.3 Third-party telnet client (Windows on Windows) 11.2.4 Third-party SSH client (standard Windows and Windows on Windows)

Many of the tasks in this section involve editing the Windows registry. As an alternative to editing the registry directly, you can create a . reg file that can be run on each user's system. For example . reg files, see 11.4 Example files for changing the Windows registry.

For details about the tasks described in this section, see the following Microsoft articles:

- Installing the Microsoft-provided telnet client: http://technet.microsoft.com/en-us/library/cc771275%28WS.10%29.aspx
- Introduction to the Windows registry: http://support.microsoft.com/kb/256986
- Backing up and restoring the Windows registry: http://support.microsoft.com/kb/322756

11.2.1 Windows operating system-provided telnet client

This procedure applies to the following cases:

- 32-bit Internet Explorer on a 32-bit operating system
- 32-bit Firefox on a 32-bit operating system
- 64-bit Internet Explorer on a 64-bit operating system
- 64-bit Microsoft Edge on a 64-bit operating system

To configure the telnet client provided by the operating system for use by a Web browser, follow these steps:

1. Windows 7, Windows 10, Windows Server 2012, Windows Server 2016 or Windows server 2019 only: Install the operating system's telnet client on the computer by following the steps appropriate to the operating system.

Windows 7 or Windows 10:

a. In the Control Panel, click Programs, and then click Programs and Features.

b. Under Tasks, click Turn Windows features on or off.

c. In the Windows Features dialog box, select the Telnet Client check box, and then click OK.

Windows Server 2012, or Windows Server 2016 or Windows Server 2019:

a. In the Server Manager, under Dashboard, click Add roles and features.

b. In the Add Roles and Features Wizard, select the Telnet Client check box, click Next, and then click Install.

2. Internet Explorer only: Enable Internet Explorer to use the telnet protocol.

a. Back up the Windows registry.

b. Use the Windows registry editor to add the [HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft \Internet Explorer\MAIN\FeatureControl\FEATURE_DISABLE_TELNET_PROTOCOL] key with the following values:

^{11.} Configuring the Telnet and SSH Protocols for Use by NNMi

Name	Туре	Data
iexplore.exe	REG_DWORD	0

3. Set file association for the URL:Telnet protocol file type.

a. Back up the Windows registry.

b. Use the Windows registry editor to modify the [HKEY_CLASSES_ROOT\telnet\shell\open \command] key with the following values:

Name	Туре	Data
(default)	REG_SZ	rundll32.exeurl.dll,TelnetProtocolHandler%l

\$1 (with a lowercase *L*) is the argument passed to telnet, usually an IP address or the fully-qualified domain name of a node.

For tighter control, you can encode the paths to the binaries in the key (as a single line). For example:

```
"C:\Windows\system32\rundll32.exe"
"C:\Windows\system32\url.dll",TelnetProtocolHandler %1
```

4. Restart the Web browser, and then, in the browser address bar, enter the telnet command:

telnet://node

node is the IP address or fully-qualified domain name of a node that runs the telnet server. If you are prompted with a security warning, permit the action. In Firefox, select the **Remember my choice for telnet links** check box.

11.2.2 Third-party telnet client (standard Windows)

This procedure applies to the following cases:

- 32-bit Internet Explorer on a 32-bit operating system
- 32-bit Internet Explorer on a 64-bit Windows 7 or Windows 10 operating system
- 32-bit Firefox on a 32-bit operating system
- 64-bit Internet Explorer on a 64-bit operating system
- 64-bit Microsoft Edge on a 64-bit operating system

To configure a third-party telnet client for use by a Web browser, follow these steps:

1. Obtain and install a third-party telnet client.

This procedure gives examples for the PuTTY client installed to C:\Program Files\PuTTY\putty.exe. The PuTTY client is available from the following Website:

http://www.putty.org

2. Internet Explorer only: Enable Internet Explorer to use the telnet protocol.

a. Back up the Windows registry.

b. Use the Windows registry editor to add the <code>[HKEY_LOCAL_MACHINE\</code>

SOFTWARE\Microsoft\Internet Explorer\MAIN\FeatureControl \FEATURE DISABLE TELNET PROTOCOL] key with the following values:

11. Configuring the Telnet and SSH Protocols for Use by NNMi

Name	Туре	Data
iexplore.exe	REG_DWORD	0

3. Set file association for the URL:Telnet protocol file type.

a. Back up the Windows registry.

b. Use the Windows registry editor to modify the [HKEY_CLASSES_ROOT\telnet\shell\open \command] key with the following values:

Name	Туре	Data
(default)	REG_SZ	"C:\Program Files\PuTTY\putty.exe" %l

\$1 (with a lowercase *L*) is the argument passed to telnet, usually an IP address or the fully-qualified domain name of a node.

In a .reg file, escape each quotation mark (") and backslash ($\)$ character with a backslash ($\)$ character.

4. Restart the Web browser, and then, in the browser address bar, enter the telnet command:

telnet://node

node is the IP address or fully-qualified domain name of a node that runs the telnet server.

If you are prompted with a security warning, permit the action.

In Firefox, select the **Remember my choice for telnet links** check box.

11.2.3 Third-party telnet client (Windows on Windows)

This procedure applies to the following cases:

- 32-bit Internet Explorer on a 64-bit operating system (other than Windows 7 or Windows 10)
- 32-bit Firefox on a 64-bit operating system

To configure a third-party telnet client for use by a Web browser, follow these steps:

1. Obtain and install a third-party telnet client.

This procedure gives examples for the PuTTY client installed to C:\Program Files\PuTTY\putty.exe. The PuTTY client is available from the following Website:

http://www.putty.org

2. Internet Explorer only: Enable Internet Explorer to use the telnet protocol.

a. Back up the Windows registry.

b. Use the Windows registry editor to add the [HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node \Microsoft\Internet Explorer\MAIN\FeatureControl

\FEATURE_DISABLE_TELNET_PROTOCOL] key with the following values:

Name	Туре	Data
iexplore.exe	REG_DWORD	0

3. Set file association for the URL:Telnet protocol file type.

a. Back up the Windows registry.

b. Use the Windows registry editor to modify the [HKEY_CLASSES_ROOT\Wow6432Node\telnet\shell \open\command] key with the following values:

^{11.} Configuring the Telnet and SSH Protocols for Use by NNMi

Name	Туре	Data
(default)	REG_SZ	"C:\Program Files\PuTTY\putty.exe" %l

l (with a lowercase *L*) is the argument passed to telnet, usually an IP address or the fully-qualified domain name of a node.

In a .reg file, escape each quotation mark (") and backslash ($\)$ character with a backslash ($\)$ character.

4. Restart the Web browser, and then, in the browser address bar, enter the telnet command:

```
telnet://node
```

node is the IP address or fully-qualified domain name of a node that runs the telnet server.

If you are prompted with a security warning, permit the action.

In Firefox, select the **Remember my choice for telnet links** check box.

11.2.4 Third-party SSH client (standard Windows and Windows on Windows)

This procedure applies to the following cases:

- 32-bit Internet Explorer on a 32-bit or 64-bit operating system
- 32-bit Firefox on a 32-bit or 64-bit operating system
- 64-bit Internet Explorer on a 64-bit operating system
- 64-bit Microsoft Edge on a 64-bit operating system

To configure a third-party SSH client for use by a Web browser, follow these steps:

1. Obtain and install a third-party SSH client.

This procedure gives examples for the PuTTY client installed to C:\Program Files\PuTTY\putty.exe.

Because PuTTY cannot correctly parse the ssh://*node* input, this example includes a script that strips the ssh:// from the input argument. The script C:\Program Files\PuTTY\ssh.js contains the following commands:

```
host = WScript.Arguments(0).replace(/ssh:/,"").replace(/\//g,"");
shell = WScript.CreateObject("WScript.Shell");
shell.Run("\"c:\\Program Files\\PuTTY\\putty.exe\" -ssh " + host);
```

This script was created for this example and is not included with PuTTY.

2. Define the SSH protocol.

a. Back up the Windows registry.

b. Use the Windows registry editor to add the [HKEY_CLASSES_ROOT\ssh] key with the following values:

Name	Туре	Data
(default)	REG_SZ	URL:SSH protocol
EditFlags	REG_DWORD	2
FriendlyTypeName	REG_SZ	SSH
URL protocol	REG_SZ	No value

3. Set file association for the URL:SSH protocol file type.

a. Back up the Windows registry.

^{11.} Configuring the Telnet and SSH Protocols for Use by NNMi

b. Use the Windows registry editor to modify the [HKEY_CLASSES_ROOT\ssh\shell\open\command] key with the following values:

Name	Туре	Data
(default)	REG_SZ	"C:\Windows\System32\WScript.exe" "C:\Program Files \PuTTY\ssh.js"%1

l (with a lowercase *L*) is the complete ssh argument, including the protocol specification. The ssh.js script passes the SSH target to PuTTY.

In a .reg file, escape each quotation mark (") and backslash ($\)$ character with a backslash ($\)$ character.

4. Restart the Web browser, and then, in the browser address bar, enter the ssh command:

ssh://*node*

node is the IP address or fully-qualified domain name of a node that runs the telnet server.

If you are prompted with a security warning, permit the action.

In Firefox, select the **Remember my choice for ssh links** check box.

^{11.} Configuring the Telnet and SSH Protocols for Use by NNMi

11.3 Configure Firefox to use telnet or SSH on Linux

On the Linux operating system, define the telnet or SSH protocol, and then configure Firefox to use the new protocol.

To complete any of the procedures in this section, you must have administrative privileges on the computer. For details, see http://kb.mozillazine.org/Register protocol.

11.3.1 Configuring Firefox to use Telnet on Linux

To configure Firefox on the Linux operating system to use the telnet protocol, follow these steps:

1. Define the telnet protocol.

a. Create the /usr/local/bin/nnmtelnet file with the following contents:

```
#!/bin/bash
#
# Linux shell script called by Firefox in response to
# telnet:// URLs for the NNMi telnet menu.
#
address=`echo $1 | cut -d : -f 2 | sed 's;/;;g'`
port=`echo $1 | cut -d : -f 3`
exec /usr/bin/xterm -e telnet $address $port
```

b. Set the script permissions to be executable by everyone:

```
chmod 755 /usr/local/bin/nnmtelnet
```

2. Configure Firefox preferences for telnet.

a. In the Firefox address bar, enter about:config.

- b. Right-click in the preference list, click New, and then click Boolean.
- c. Enter the preference name network.protocol-handler.expose.telnet.
- d. Select the preference value false.
- 3. Configure Firefox to use the newly-defined protocol.

a. Browse to a telnet link.

You can create a simple HTML file containing the link, or you can use the Actions > Node Access > Telnet... (from client) menu item in the NNMi console. Typing the link directly into the address bar does not have the same effect.

b. In the Launch Application window, click Choose, and then select /usr/local/bin/nnmtelnet.

c. Select the Remember my choice for telnet links check box.

11.3.2 Configuring Firefox to use SSH on Linux

To configure Firefox on the Linux operating system to use the SSH protocol, follow these steps:

1. Define the SSH protocol.

a. Create the /usr/local/bin/nnmssh file with the following contents:

```
#!/bin/bash
#
# Linux shell script called by Firefox in response to
# ssh:// URLs for the NNMi SSH menu.
#
```

^{11.} Configuring the Telnet and SSH Protocols for Use by NNMi

```
address=`echo $1 | cut -d : -f 2 | sed 's;/;;g'`
port=`echo $1 | cut -d : -f 3`
exec /usr/bin/xterm -e ssh $address $port
```

b. Set the script permissions to be executable by everyone:

```
chmod 755 /usr/local/bin/nnmssh
```

2. Configure Firefox preferences for SSH.

a. In the Firefox address bar, enter about:config.

- **b.** Right-click in the preference list, click **New**, and then click **Boolean**.
- c. Enter the preference name network.protocol-handler.expose.ssh.
- $\textbf{d.} \ Select \ the \ preference \ value \ \texttt{false}.$
- 3. Configure Firefox to use the newly-defined protocol.

a. Browse to an SSH link.

You can create a simple HTML file containing the link, or you can use the new SSH menu item that you defined in the NNMi console. Typing the link directly into the address bar does not have the same effect.

b. In the Launch Application window, click Choose, and then select /usr/local/bin/nnmssh.

c. Select the Remember my choice for ssh links check box.

^{11.} Configuring the Telnet and SSH Protocols for Use by NNMi

11.4 Example files for changing the Windows registry

If many NNMi users need to use the telnet or SSH protocol to access managed nodes from the NNMi console, you might be able to automate the Windows registry updates with one or more .reg files. This section contains example .reg files that you can use as a basis for creating your own .reg files. Note that the registry keys are located in a different path for running 32-bit applications on 64-bit versions of Windows than they are when the application and operating system match.

For details, see the Microsoft article at http://support.microsoft.com/kb/310516.

11.4.1 Example nnmtelnet.reg

This registry content example applies to 11.2.1 Windows operating system-provided telnet client.

```
Windows Registry Editor Version 5.00
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Internet Explorer\MAIN\FeatureControl\FEATURE_
DISABLE_TELNET_PROTOCOL]
"iexplore.exe"=dword:00000000
[HKEY_CLASSES_ROOT\telnet\shell\open\command]
@="\"C:\\Windows\\system32\\rundll32.exe\"
\"C:\\Windows\\system32\\url.dll\",TelnetProtocolHandler %1"
```

11.4.2 Example nnmputtytelnet.reg

This registry content example applies to 11.2.2 Third-party telnet client (standard Windows).

```
Windows Registry Editor Version 5.00
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Internet Explorer\MAIN\FeatureControl\FEATURE_
DISABLE_TELNET_PROTOCOL]
"iexplore.exe"=dword:0c0000000
[HKEY_CLASSES_ROOT\telnet\shell\open\command]
@="\"C:\\Program Files\\PuTTY\\putty.exe\" %1"
```

11.4.3 Example nnmtelnet32on64.reg

This registry content example applies to 11.2.3 Third-party telnet client (Windows on Windows).

```
Windows Registry Editor Version 5.00
[HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Microsoft\Internet Explorer\MAIN\FeatureCont
rol\FEATURE_DISABLE_TELNET_PROTOCOL]
"iexplore.exe"=dword:0000000
[HKEY_CLASSES_ROOT\Wow6432Node\telnet\shell\open\command]
@="\"C:\\Program Files\\PuTTY\\putty.exe\" %1"
```

11.4.4 Example nnmssh.reg

This registry content example applies to 11.2.4 Third-party SSH client (standard Windows and Windows on Windows).

^{11.} Configuring the Telnet and SSH Protocols for Use by NNMi
Windows Registry Editor Version 5.00
[HKEY_CLASSES_ROOT\ssh]
@="URL:ssh Protocol"
"EditFlags"=dword:00000002
"FriendlyTypeName"="Secure Shell"
"URL Protocol"=""
[HKEY_CLASSES_ROOT\ssh\shell\open\command] @="\"C:\\Windows\\System32\\WScript.exe\"
\"c:\\Program_Files\\PuTTY\\ssh.js\" %1"

^{11.} Configuring the Telnet and SSH Protocols for Use by NNMi



Integrating NNMi with a Directory Service Through LDAP

This chapter explains how to consolidate the storage of user names, passwords, and, optionally, NNMi user group assignments by integrating NNMi with a directory service.



Together, the following items define an NNMi user:

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

Important

If your device is SNMPv1 or SNMPv2c, note the following:

SNMPv1 and SNMPv2c send their information packets in clear text.

To secure your environment, use SNMPv3 or add protections, such as firewall controls, for the flow of SNMP traps and the collection of information from your devices.

NNMi provides several options for where the NNMi user access information is stored.

The following table shows the databases that store the NNMi user access information for each configuration mode.

Table 12-1: Options for storing user information

Mode	User account	User group	User group membership
Internal (option 1)	NNMi	NNMi	NNMi
Mixed (option 2)	Mixed (account name in NNMi, account password in LDAP)	NNMi	NNMi
External (option 3)	Directory service	Both	Directory service

^{12.} Integrating NNMi with a Directory Service Through LDAP

NNMi uses the Lightweight Directory Access Protocol (LDAP) to communicate with the directory service. One of the following modes shown in the table above must be used in order to use LDAP with NNMi:

• Mixed Mode (referred to originally as Option 2): Some NNMi user information is in the NNMi database and some NNMi user information is in a directory service

When you use the mixed mode, you configure NNMi to store user names, user groups, and user group mappings in the NNMi database, and you rely on a directory service for user names and passwords (user account definitions). This means that account name information must be stored in both NNMi and LDAP. Account passwords, on the other hand, are stored only in LDAP.

• External Mode (referred to originally as Option 3): All NNMi user information is in the directory service When you use the external mode, there is no need to add user account information to NNMi, because you use LDAP to store all user account information.

To add new user accounts or to modify existing accounts when you use the mixed mode, you must select the **Directory Service Account** check box. When you are configuring user accounts, you must not select the **Directory Service Account** check box for some users and not select it for others (in effect, combining the internal, mixed, and external modes); doing so will result in an unsupported configuration.

12.1.1 Internal mode: Storing all NNMi user information in the NNMi database

With configuration using the internal mode, NNMi accesses the NNMi database for all user access information, which the NNMi administrator defines and maintains in the NNMi console. The user access information is local to NNMi. NNMi does not access a directory service, and NNMi is not configured to retrieve information from the LDAP configuration file.

The figure below shows the information flow for this option, which is appropriate in the following situations:

- The number of NNMi users is small.
- No directory service is used.

For details about setting up all user information in the NNMi database, see *NNMi Configuration Settings to Control NNMi Access* in NNMi Help. You do not need to read the present chapter.

Figure 12-1: Flow of NNMi user sign-in information in the internal mode



12.1.2 Mixed mode: Storing some NNMi user information in the NNMi database and some NNMi user information in a directory service

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

- The directory service administrator maintains the user names and passwords in the directory service.
- The NNMi administrator enters the user names (as defined in the directory service), user group definitions, and the user group mappings in the NNMi console.
- The NNMi administrator configures NNMi's LDAP configuration file to describe to NNMi the directory service database schema for user names.

In the figure below, making the last line a comment line prevents NNMi from acquiring NNMi user group information from the directory service.

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





The figure above shows the information flow for this mode, which is appropriate in the following situations:

- The number of NNMi users is small, and a directory service is available.
- The NNMi administrator wants to manage the user groups instead of requiring a directory service change for each user group change.
- The directory service group definitions are available.

For details about integrating with a directory service for the user name and password, see the rest of this chapter and *Lightweight Directory Access Protocol (LDAP) to Control NNMi Access* in NNMi Help.

12.1.3 External mode: Storing all NNMi user information in a directory service

NNMi accesses a directory service for all user access information, which is defined externally to NNMi and is available to other applications. The NNMi user groups to which a user belongs is determined by the user's membership in one or more directory service groups.

The configuration and maintenance of NNMi user access information is a joint effort as described here:

- The directory service administrator maintains the user names, passwords, and group membership in the directory service.
- The NNMi administrator maps the directory service groups to NNMi user groups in the NNMi console.
- The NNMi administrator configures NNMi's LDAP configuration file to describe to NNMi the directory service database schema for user names and groups.

The figure below shows the information flow for this mode, which is appropriate for environments where the directory service can be modified to include user groups that align with the people who need access to NNMi.

^{12.} Integrating NNMi with a Directory Service Through LDAP





Because this option is an expansion of the mixed mode scenario, we recommend the following configuration process:

- 1. Configure and verify NNMi user name and password retrieval from the directory service.
- 2. Configure NNMi user group retrieval from the directory service.

For details about integrating with a directory service that stores all user information, see the rest of this chapter and *Lightweight Directory Access Protocol (LDAP) to Control NNMi Access* in NNMi Help.

^{12.} Integrating NNMi with a Directory Service Through LDAP

12.2 Configuring NNMi to access a directory service

You can configure directory service access in the nms-auth-config.xml file.

The file is located at:

- Windows: %NnmDataDir%nmsas\NNM\conf
- Linux: \$NnmDataDir/nmsas/NNM/conf

By default, the nms-auth-config.xml file available in this location does not contain the XML elements required for LDAP configuration.

You can manually add all the necessary XML elements to this file by following the instructions in this section.

NNMi places a sample nms-auth-config.xml file in a different location, which can be used for reference.

The sample nms-auth-config.xml file is available in the following location:

- Windows: %NnmInstallDir%newconfig\HPOvNnmAS\nmsas\conf
- Linux: \$NnmInstallDir/newconfig/HPOvNnmAS/nmsas/conf

🛛 Тір

You can also copy the entire <ldapLogin> element from the sample nms-auth-config.xml file,and then make necessary modifications.

For details about the nms-auth-config.xml file, see 12.6 LDAP configuration file reference.

For details about the general structure of a directory service, see 12.3 Directory service queries.

To configure access for the mixed mode, complete the following tasks:

- 12.2.1 Task 1: Back up the current NNMi user information
- 12.2.2 Task 2: (Optional) Configure secure communications to the directory service
- 12.2.3 Task 3: Configure user access from the directory service
- 12.2.4 Task 4: Test the user name and password configuration
- 12.2.9 Task 9: Clean up to prevent unexpected access to NNMi
- 12.2.10 Task 10: (Optional) Map the user groups to security groups

To configure access for the external mode, complete the following tasks:

- 12.2.1 Task 1: Back up the current NNMi user information
- 12.2.2 Task 2: (Optional) Configure secure communications to the directory service
- 12.2.3 Task 3: Configure user access from the directory service
- 12.2.4 Task 4: Test the user name and password configuration
- 12.2.5 Task 5: (Configuring for the external mode only) Configure group retrieval from the directory service

🖹 Note

If you plan to store NNMi user groups in the directory service, the directory service must be configured with the NNMi user groups. For details, see 12.4 Directory service configuration for storing NNMi user groups.

- 12.2.6 Task 6: (Configuring for the external mode only) Map the directory service groups to NNMi user groups
- 12.2.7 Task 7: (Configuring for the external mode only) Test the NNMi user group configuration
- 12.2.8 Task 8: (Configuring for the external mode only) Configure NNMi user groups for incident assignment
- 12.2.9 Task 9: Clean up to prevent unexpected access to NNMi
- 12.2.10 Task 10: (Optional) Map the user groups to security groups

12.2.1 Task 1: Back up the current NNMi user information

Back up the user information in the NNMi database:

nnmconfigexport.ovpl -c account -u user -p password -f NNMi_database_accounts.xml

12.2.2 Task 2: (Optional) Configure secure communications to the directory service

If the directory service requires the use of secure sockets layer (SSL), import your company's certificate into the NNMi truststore as described in 10.3.8 Configuring an SSL connection to the Directory service.

12.2.3 Task 3: Configure user access from the directory service

Complete this task for mixed mode and external mode only. Follow the appropriate procedure for your directory service.

1. Go to the following directory:

- Windows: %NnmDataDir%nmsas\NNM\conf
- Linux: \$NnmDataDir/nmsas/NNM/conf
- 2. Back up the nms-auth-config.xml file that was shipped with NNMi, and then open the file in any text editor.
- 3. Specify values for the following elements:



NNMi places a sample nms-auth-config.xml file in a different location, which can be used for reference.

The sample nms-auth-config.xml file is available in the following location:

- Windows: %NnmInstallDir%newconfig\HPOvNnmAS\nmsas\conf
- Linux: \$NnmInstallDir/newconfig/HPOvNnmAS/nmsas/conf

^{12.} Integrating NNMi with a Directory Service Through LDAP

You can also copy the entire <ldapLogin> element from the sample nms-auth-config.xml file, and then make necessary modifications.

<enabled> </enabled>		Specify true to use the nms-auth-config.xml file. By default, this element is set to false.	
<userrolef <td>ilterList> FilterList></td><td>Specify the NNMi roles to which NNMi users can assign incidents. To assign incidents to all operators, administrators, and guests, add this: <userrolefilterlist> admin guest level2 level1 </userrolefilterlist></td></userrolef 	ilterList> FilterList>	Specify the NNMi roles to which NNMi users can assign incidents. To assign incidents to all operators, administrators, and guests, add this: <userrolefilterlist> admin guest level2 level1 </userrolefilterlist>	
<connecttin <td>meLimit> imeLimit></td><td>Specify the connection timeout value in milliseconds. The default value is 10000 (10 seconds). If you are encountering timeouts during NNMi user sign in, increase this value. For example: <connecttimelimit>10000</connecttimelimit></td></connecttin 	meLimit> imeLimit>	Specify the connection timeout value in milliseconds. The default value is 10000 (10 seconds). If you are encountering timeouts during NNMi user sign in, increase this value. For example: <connecttimelimit>10000</connecttimelimit>	
<searchtim <td>eLimit> neLimit></td><td>Specify the search timeout value in milliseconds. The default value is 30000 (30 seconds). If you are encountering timeouts during NNMi user sign in, increase this value. For example: <searchtimelimit>30000</searchtimelimit></td></searchtim 	eLimit> neLimit>	Specify the search timeout value in milliseconds. The default value is 30000 (30 seconds). If you are encountering timeouts during NNMi user sign in, increase this value. For example: <searchtimelimit>30000</searchtimelimit>	
<server></server>		Container element to contain all LDAP configuration information.	
	<host> </host>	<pre>URL of the LDAP server with port. For example: • To use HTTP:</pre>	
	<secure> </secure>	Specify true if you want to use HTTPS. Otherwise, specify false.	
Note: The s LDAP serve specified at	erver element can be specified multiple er configuration. In such cases, attempts t the top.	times if the same information can be acquired from multiple servers in a redundant to establish a connection will be made in order starting from the connection destination	
<bindcrede< td=""><td>ential></td><td>Container element to include bind credentials (mandatory for directory services that do not support anonymous logon).</td></bindcrede<>	ential>	Container element to include bind credentials (mandatory for directory services that do not support anonymous logon).	
	<binddn> </binddn>	Specify the bind DN.	
	 <bindcredential> </bindcredential>	Specify the bind DN password in the encrypted format. Run the "nnmldap.ovpl -encrypt <mypassword>" command to encrypt the password.</mypassword>	
<td>ential></td> <td></td>	ential>		
<users></users>		Container element to include all user configuration details.	
	<usersearch></usersearch>	Container element to include the configuration information for searching users. Specify the <usersearch></usersearch> setting only once. Specifying this setting more than once is not supported.	
	<base/>	For example:	

T 1 1	100	– 1	C 11		• •	• •	r		C .	
lable	12-2:	Elements	or the	IdapLo	sain :	Section	ot nm:	s-autn-o	contia.	xmi

12. Integrating NNMi with a Directory Service Through LDAP

		 For Active Directory: <base/> CN={0} For other LDAP technologies: <base/> SAMAccountName={0}
	<basecontextdn></basecontextdn>	<pre>For Active Directory, specify the portion of the directory service domain that stores user records. For example: For Active Directory: OU=Users,OU=Accounts,DC=mycompany,DC=com For other LDAP technologies: ou=People,o=example.com</pre>
Note: In mi	xed mode, specify the <rolesearch></rolesearch>	setting only once, as follows, and then perform step 4 below.

```
<roleSearch>
<roleBase></roleBase>
<roleContextDN></roleContextDN>
</roleSearch>
```

- 4. After editing the nms-auth-config.xml file (%NnmDataDir%nmsas\NNM\conf (Windows) or \$NnmDataDir/nmsas/NNM/conf (Linux)), run the following command:
 - Windows:

```
%NnmInstallDir%bin\nnmldap.ovpl -reload
```

• Linux:

```
$NnmInstallDir/bin/nnmldap.ovpl -reload
```

12.2.4 Task 4: Test the user name and password configuration

1. In the LDAP configuration file, set defaultRole to guest for testing purposes.

You can change this value at any time.

```
In nms-auth-config.xml file add the following content before the usersearch element:
    <defaultRoles>
    <role>guest</role>
```

</defaultRoles>

- 2. Save the LDAP configuration file.
- 3. Force NNMi to re-read the LDAP configuration file by executing the following command: nnmldap.ovpl-reload
- 4. Sign in to the NNMi console with a user name and password that are defined in the directory service. Run this test with a user name that is not already defined in the NNMi database.
- 5. Verify the user name and NNMi role (Guest) in the title bar of the NNMi console.
 - If user sign-in works correctly, go to step 8 of this task.
 - If user sign-in does not work correctly, continue on to step 6.
 - After each test, sign out of the NNMi console to clear the session credentials.
- 6. Test the configuration for one user by running the following command:

^{12.} Integrating NNMi with a Directory Service Through LDAP

Replace NNMi-user with the sign-in name of an NNMi user as defined in the directory service.

Examine the command output and respond appropriately. The following are suggestions:

- Verify that you completed correctly 12.2.3 Task 3: Configure user access from the directory service.
- Follow the detailed configuration process in 12.3.4 User identification.

Note

In mixed mode, the following message is output. However, this does not indicate a problem in operation, because the LDAP group is not referenced in mixed mode. Therefore, you can ignore the message.

7. Repeat steps 1 through 5 until you see the expected result when signing in to the NNMi console.

8. After you can sign in, choose your strategy:

- If you plan to store NNMi user group membership in the NNMi database (configuring for the mixed mode), go to 12.2.9 Task 9: Clean up to prevent unexpected access to NNMi.
- If you plan to store NNMi user group membership in the directory service (configuring for the external mode), continue on to Task 5.

12.2.5 Task 5: (Configuring for the external mode only) Configure group retrieval from the directory service

Complete this task to configure group retrieval for the external mode. Follow the appropriate procedure for your directory service.

1. Go to the following directory:

- Windows: %NnmDataDir%nmsas\NNM\conf
- Linux: \$NnmDataDir/nmsas/NNM/conf
- 2. Take a backup of the nms-auth-config.xml file, and then open the file with a text editor.

3. Modify the following elements:



NNMi places a sample nms-auth-config.xml file in a different location, which can be used for reference.

The sample nms-auth-config.xml file is available in the following location:

• Windows: %NnmInstallDir%newconfig\HPOvNnmAS\nmsas\conf

12. Integrating NNMi with a Directory Service Through LDAP

• Linux: \$NnmInstallDir/newconfig/HPOvNnmAS/nmsas/conf

You can also copy the entire <ldapLogin> element from the sample nms-auth-config.xml file, and then make necessary modifications.

Table 12-3: Elements of the IdapLogin Section of nms-auth-config.xml

<rolesearc< td=""><td>h></td><td>Placeholder element to include the user role information. Specify the <rolesearch></rolesearch> setting only once. You cannot specify this setting more than once.</td></rolesearc<>	h>	Placeholder element to include the user role information. Specify the <rolesearch></rolesearch> setting only once. You cannot specify this setting more than once.
	<rolebase>member= {1} </rolebase>	Replace member with the name of the group attribute that stores the directory service user ID in the directory service domain.
	<rolecontextdn> </rolecontextdn>	 Specify the portion of the directory service domain that stores group records. The format is a comma-separated list of directory service attribute names and values. For example: For Active Directory CN=Users, DC=ldapserver, DC=mycompany, DC=com
		• For other LDAP technologies ou=Groups,o=example.com
<td>ch></td> <td></td>	ch>	

4. Save the file.

5. Run the following command:

```
nnmldap.ovpl -reload
```

12.2.6 Task 6: (Configuring for the external mode only) Map the directory service groups to NNMi user groups

1. In the NNMi console, map the predefined NNMi user groups to their counterparts in the directory service:

a. Open the User Groups view.

In the Configuration workspace, expand Security, and then click User Groups.

- b. Double-click the **admin** row.
- c. In the **Directory Service Name** field, enter the full distinguished name of the directory service group for NNMi administrators.
- d. Click Save and Close.
- e. Repeat steps b through d for each of the guest, level1, and level2 rows.

These mappings provide NNMi console access. Every user who will access the NNMi console must be in a directory service group that is mapped to one of the predefined NNMi user groups named in this step.

- 2. For other groups containing one or more NNMi users in the directory service, create a new user group in the NNMi console.
 - a. Open the User Groups view.

In the Configuration workspace, expand Security, and then click User Groups.

- b. Click New, and then enter the information for the group:
 - Set Name to any unique value. Short names are recommended.
 - Set Display Name to the value users see.

^{12.} Integrating NNMi with a Directory Service Through LDAP

- Set Directory Service Name to the full distinguished name of the directory service group.
- Set **Description** to text that describes the purpose of this NNMi user group.
- c. Click Save and Close.
- d. Repeat step b and step c for each additional directory service group of NNMi users.

These mappings provide topology object access in the NNMi console. Each directory service group can be mapped to multiple NNMi user groups.

12.2.7 Task 7: (Configuring for the external mode only) Test the NNMi user group configuration

- 1. Save NNMi's LDAP configuration file (nms-auth-config.xml).
- 2. Force NNMi to re-read the LDAP configuration file by executing the following command:

nnmldap.ovpl -reload

- 3. Sign in to the NNMi console with a user name and password that are defined in the directory service. Run this test with a user name that is not already defined in the NNMi database and is a member of a directory service group that is mapped to the admin, level1, or level2 NNMi user group.
- 4. Verify the user name and NNMi role (as configured in the **Display Name** field in the **User Groups** view) in the title bar of the NNMi console.
 - If user sign-in works correctly, go to step 8.
 - If user sign-in does not work correctly, continue on to step 5.

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

5. Test the configuration for one user by running the following command:

nnmldap.ovpl -diagnose NNMi-user

Replace NNMi-user with the sign-in name of an NNMi user as defined in the directory service.

Examine the command output and respond appropriately. The following are suggestions:

- Verify that you completed correctly 12.2.5 Task 5: (Configuring for the external mode only) Configure group retrieval from the directory service.
- Verify for each predefined NNMi user group that you completed correctly 12.2.6 Task 6: (Configuring for the external mode only) Map the directory service groups to NNMi user groups.
- Follow the detailed configuration process in 12.3.5 User group identification.
- 6. Repeat steps 1 through 4 until you see the expected result when signing in to the NNMi console.

12.2.8 Task 8: (Configuring for the external mode only) Configure NNMi user groups for incident assignment

- 1. Sign in to the NNMi console with a user name and password that are defined in the directory service.
- 2. In any incident view, select an incident, and then click Actions > Assign > Assign Incident. Verify that you can assign the incident to a user in each of the NNMi roles specified by the userRoleFilterList parameter.

^{12.} Integrating NNMi with a Directory Service Through LDAP

12.2.9 Task 9: Clean up to prevent unexpected access to NNMi

1. (Optional) Either change the value of or comment out the defaultRole parameter in the LDAP configuration file.

If you change the value of, or comment out, the defaultRole element or parameter in the LDAP configuration file, force NNMi to re-read the LDAP configuration file by running the following command:

nnmldap.ovpl -reload

- 2. (Configuring for the mixed mode only) To store user group membership in the NNMi database, reset the user access information in the NNMi database as follows:
 - a. Remove any pre-existing user access information. (Delete all rows in the User Accounts view.) For details, see *Delete a User Account* in NNMi Help.
 - b. For each NNMi user, create a new object in the User Accounts view for the user name.
 - For the Name field, enter the user name as defined in the directory service.
 - Select the Directory Service Account check box.
 - Do not specify a password.

For details, see User Account Tasks in NNMi Help.

- c. For each NNMi user, map the user account to one or more NNMi user groups. For details, see *Map User Accounts to User Groups (User Account Mapping Form)* in NNMi Help.
- d. Update incident ownership so that each assigned incident is associated with a valid user name. For details, see *Manage Incident Assignments* in NNMi Help.
- 3. (Configuring for the external mode only) To rely on the directory service for user group membership, reset the user access information in the NNMi database as follows:
 - a. Remove any pre-existing user access information. (Delete all rows in the User Accounts view.) For details, see *Delete a User Account* in NNMi Help.
 - b. Update incident ownership so that each assigned incident is associated with a valid user name. For details, see *Manage Incident Assignments* in NNMi Help.

12.2.10 Task 10: (Optional) Map the user groups to security groups

For details, see Security Group Mapping Tasks in NNMi Help.

^{12.} Integrating NNMi with a Directory Service Through LDAP

12.3 Directory service queries

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

12.3.1 Directory service access

LDAP queries to a directory service use the following format:

ldap://directory-service-host:port/search-string

- ldap is the protocol indicator. Use this indicator for both standard connections and SSL connections to the directory service.
- *directory-service-host* is the fully-qualified name of the computer that hosts the directory service.
- *port* is the port that the directory service uses for LDAP communication. The default port for non-SSL connections is 389. The default port for SSL connections is 636.
- search-string contains the information request. For details, see 12.3.2 Directory service content and RFC 1959, An LDAP URL Format, which is available at:

http://www.ietf.org/rfc/rfc1959.txt

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

If the directory service (for example, Active Directory) does not permit anonymous access, the directory service denies LDAP queries from a Web browser. In this case, you can use a third-party LDAP browser (for example, the LDAP browser included in Apache Directory Studio) to validate your configuration parameters.

12.3.2 Directory service content

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

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

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

^{12.} Integrating NNMi with a Directory Service Through LDAP

Example content structure for Active Directory

In this example, the following items are of interest:

• The distinguished name of the user John Doe is:

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

• The distinguished name of the group USERS-NNMi-Admin is:

```
CN=USERS-NNMi-Admin,OU=Groups,OU=Accounts,DC=example,DC=com
```

• The group attribute that stores the directory service user ID is: member

Example LDIF file excerpt:

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

The following figure illustrates this directory service domain.

^{12.} Integrating NNMi with a Directory Service Through LDAP

Figure 12-4: Example domain for Active Directory



Example content structure for other directory services

In this example, the following items are of interest:

• The distinguished name of the user John Doe is:

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

• The distinguished name of the group USERS-NNMi-Admin is:

cn=USERS-NNMi-Admin,ou=Groups,o=example.com

• The group attribute that stores the directory service user ID is: member

Example LDIF file excerpt:

```
groups |USERS-NNMi-Admin
dn: cn=USERS-NNMi-Admin,ou=Groups,o=example.com
cn: USERS-NNMi-Admin
```





12.3.3 Information owned by the directory service administrator

Table 12-4: Information for retrieving user names and passwords from a directory service and Table 12-5: Information for retrieving group membership from a directory service list the information to obtain from the directory service administrator before configuring NNMi for LDAP access to a directory service.

- If you plan to use the directory service for user names and passwords only (configuring for the mixed mode), gather the information shown in Table 12-4: Information for retrieving user names and passwords from a directory service.
- If you plan to use the directory service for all NNMi access information (configuring for the external mode), gather the information shown in both Table 12-4: Information for retrieving user names and passwords from a directory service and Table 12-5: Information for retrieving group membership from a directory service.

^{12.} Integrating NNMi with a Directory Service Through LDAP

Table 12-4: Information for retrieving user names and passwords from a directory service

Information	Active Directory example	Other directory services example			
The fully-qualified name of the computer that hosts the directory service	directory_service_host.example.com				
The port that the directory service uses for LDAP communication	 389 for non-SSL connections 636 for SSL connections				
Does the directory service require an SSL connection?	If yes, obtain a copy of your company's truststore certificate and see 10.3.8 Configuring SSL connection to the Directory service.				
The distinguished name for one user name that is stored in the directory service (to demonstrate the directory service domain)	CN=john.doe@example.com, OU=User s,OU=Accounts, DC=example,DC=com	<pre>uid=john.doe@example.com, ou=Pe ople,o=example.com</pre>			

Table 12-5: Information for retrieving group membership from a directory service

Information	Active Directory example	Other directory services example
The distinguished name for identifying the groups to which a user is assigned	The memberOf user attribute identifies the groups.	 ou=Groups,o=example.com cn=USERS-NNMi-*, ou=Groups,o=example.com
The method of identifying a user within a group	 CN=john.doe@example.com, OU=Users,OU=Accounts, DC=example,DC=com CN=john.doe@example.com 	 cn=john.doe@example.com, ou=People,o=example.com cn=john.doe@example.com
The group attribute that stores the directory service user ID	member	member
The names of the groups in the directory service that apply to NNMi access	 CN=USERS-NNMi-Admin, OU=Groups,OU=Accounts, DC=example,DC=com CN=USERS-NNMi-Level2, OU=Groups,OU=Accounts, DC=example,DC=com CN=USERS-NNMi-Level1, OU=Groups,OU=Accounts, DC=example,DC=com CN=USERS-NNMi-Client, OU=Groups,OU=Accounts, DC=example,DC=com CN=USERS-NNMi-Guest, OU=Groups,OU=Accounts, DC=example,DC=com 	 cn=USERS-NNMi-Admin, ou=Groups,o=example.com cn=USERS-NNMi-Level2, ou=Groups,o=example.com cn=USERS-NNMi-Level1, ou=Groups,o=example.com cn=USERS-NNMi-Client, ou=Groups,o=example.com cn=USERS-NNMi-Guest, ou=Groups,o=example.com

12.3.4 User identification

User identification applies to the mixed mode and the external mode.

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

^{12.} Integrating NNMi with a Directory Service Through LDAP

In the LDAP configuration file, the user distinguished name is the concatenation of the <base> and <baseContextDN> elements in the nms-auth-config.xml file. If the password returned by the directory service matches the sign-in password the user entered into the NNMi console, user sign in continues.

When configuring user identification for the mixed mode, the following information applies:

- For NNMi console access, NNMi examines the following information and grants the user the highest possible privileges:
 - The value of the defaultRole parameter in the LDAP configuration file
 - This user's membership in the predefined NNMi user groups in the NNMi console
- For NNMi topology object access, NNMi grants access according to the security group mappings for the NNMi user groups in the NNMi console to which this user belongs.

When configuring user identification for the external mode, the following information applies:

• For NNMi console access, NNMi examines the following information and grants the user the highest possible privileges:

- The value of the defaultRole parameter in the LDAP configuration file

- This user's membership in the directory service groups that are mapped (with the **Directory Service Name** field) to the predefined NNMi user groups in the NNMi console

• For NNMi topology object access, NNMi grants access according to the security group mappings for the groups in the directory service to which this user belongs (as mapped to NNMi user groups in the NNMi console).

Active Directory user identification example

```
If the nms-auth-config.xml file contains <base>CN={0}</
base><baseContextDN>OU=Users,OU=Accounts,DC=example,DC=com</baseContextDN>,
and a user signs in to NNMi as john.doe, the string passed to the directory service is:
CN=john.doe,OU=Users,OU=Accounts,DC=example,DC=com
```

Other directory services user identification example

```
If the nms-auth-config.xml file contains <base>uid={0}@example.com</base>contextDN>ou=People, o=example.com</baseContextDN>, and a user signs in to NNMi as john.doe, the string passed to the directory service is:
```

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

12.3.5 User group identification

User group identification applies to configuring for the external mode.

NNMi determines the user groups for an NNMi user as follows:

- 1. NNMi compares the values of the external names of all user groups configured in the NNMi console with the names of the directory service groups.
- 2. For any user group match, NNMi then determines whether the NNMi user is a member of that group in the directory service.

In the NNMi console, short text strings identify the unique names of the predefined NNMi user groups that grant NNMi console access. These text strings are also required by the defaultRole and userRoleFilterList parameters in the LDAP configuration file. The following table maps the unique names of these groups to their display names.

^{12.} Integrating NNMi with a Directory Service Through LDAP

Table 12-6: NNMi user group name mapping

NNMi role name in the NNMi console	User group unique name and text string in NNMi configuration files
Administrator	admin
Global operators	globalops
Operator level 2	level2
Operator level 1	level1
Guest	guest
Web service client	client

The NNMi global operator user group (globalops) can access only all topology objects. A user is able to access the NNMi console only if that user is assigned to another user group (admin, level2, level1, or guest).

Because the globalops user group is mapped to all security groups by default, the administrator must not map this user group to security groups.

(1) Configuring user group retrieval from the directory service (detailed approach)

If the simple approach described in 12.2.5 Task 5: (Configuring for the external mode only) Configure group retrieval from the directory service in 12.2 Configuring NNMi to access a directory service did not work correctly, follow these steps:

- 1. Obtain the required user information from the directory service administrator.
- 2. Verify the format of group names and group members in the directory service by completing the appropriate procedure:
 - LDAP browser approach for Active Directory: See (2) Determining how the directory service identifies a group and group membership (LDAP browser approach for Active Directory), below.
 - LDAP browser approach for other directory services: See (3) Determining how the directory service identifies a group and group membership (LDAP browser approach for other directory services), below.
 - Web browser approach for other directory services: See (4) Determining how the directory service identifies a group (Web browser approach), below.
- 3. Configure the LDAP configuration file.
 - a. Open the nms-auth-config.xml file in any text editor.
 - b. Set the role element to correlate user names to the way user names are stored for groups in the directory service. Replace the actual user name with one of the following expressions:
 - Use {0} to denote the user name entered for sign-in (for example, john.doe).
 - Use {1} to denote the distinguished name of the authenticated user as returned by the directory service (for example, uid=john.doe@example.com, ou=People, o=example.com).
 - c. Set the roleContextDN element to the portion of the directory service domain that stores group records. The format is a comma-separated list of directory service attribute names and values.

For example:

• For Active Directory: CN=Users, DC=ldapserver, DC=mycompany, DC=com

12. Integrating NNMi with a Directory Service Through LDAP

• For other LDAP technologies: ou=Groups,o=example.com

4. Test the configuration as described in 12.2 Configuring NNMi to access a directory service.

(2) Determining how the directory service identifies a group and group membership (LDAP browser approach for Active Directory)

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

- 1. Navigate to the portion of the directory service domain that stores user information.
- 2. Identify a user who requires access to NNMi, and then examine the format of the distinguished names for the groups associated with that user.
- 3. Navigate to the portion of the directory service domain that stores group information.
- 4. Identify the groups that correspond to NNMi user groups, and then examine the format of the names for the users associated with a group.

(3) Determining how the directory service identifies a group and group membership (LDAP browser approach for other directory services)

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

- 1. Navigate to the portion of the directory service domain that stores group information.
- 2. Identify the groups that correspond to NNMi user groups, and then examine the format of the distinguished names for those groups.
- 3. Also examine the format of the names for the users associated with a group.

(4) Determining how the directory service identifies a group (Web browser approach)

1. In a supported Web browser, enter the following URL:

ldap://directory-service-host:port/group-search-string

- *directory-service-host* is the fully-qualified name of the computer that hosts the directory service.
- port is the port that the directory service uses for LDAP communication.
- *group-search-string* is the distinguished name for a group name that is stored in the directory service (for example: cn=USERS-NNMi-Admin, ou=Groups, o=example.com).
- 2. Evaluate the results of the directory service access test.
 - If you see a message that the directory service does not contain the requested entry, verify the value of *group-search-string*, and then repeat step 1.
 - If you see the appropriate list of groups, the access information is correct.
- 3. Examine the group properties to determine the format of the names for the users associated with that group.

12.4 Directory service configuration for storing NNMi user groups

If you plan to store NNMi user groups in a directory service (configuring for the external mode), the directory service must be configured with NNMi user group information. Ideally, the directory service already contains appropriate user groups. If this is not the case, the directory service administrator can create new user groups specifically for NNMi user group assignment.

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

12. Integrating NNMi with a Directory Service Through LDAP

12.5 Troubleshooting the directory service integration

1. Verify the NNMi LDAP configuration by running the following command:

nnmldap.ovpl -info

If the reported configuration is not as expected, verify the settings in the LDAP configuration file.

2. Force NNMi to re-read the LDAP configuration file by executing the following command:

nnmldap.ovpl -reload

3. Test the configuration for one user by running the following command:

nnmldap.ovpl -diagnose NNMi-user

Replace NNMi-user with the sign-in name of an NNMi user as defined in the directory service.

Examine the command output and respond appropriately.

Note

In mixed mode, the following message is output. However, this does not indicate a problem in operation, because the LDAP group is not referenced in mixed mode. Therefore, you can ignore the message.

4. Verify that the directory service contains the expected records.

Use a Web browser or a third-party LDAP browser (for example, the LDAP browser included in Apache Directory Studio) to examine the directory service information.

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

http://www.ietf.org/rfc/rfc1959.txt

- 5. View the %NnmDataDir%log\nnm\nnm.log (Windows) or /var/opt/OV/log/nnm/nnm.log (Linux) log file to verify that the sign-in request is correct and to determine if any errors occurred:
 - A message similar to the line below indicates that the directory service requires HTTPS communication. In this case, enable SSL, as described in 10.3.8 Configuring an SSL connection to the Directory service.

```
javax.naming.AuthenticationNotSupportedException:[LDAP:error code 13 - confidentia
lity required]
```

• A message similar to the line below indicates that a timeout occurred while communicating with the directory service. In this case, increase the value of searchTimeLimit in the LDAP configuration file.

javax.naming.TimeLimitExceededException:[LDAP: error code 3 - Timelimit Exceeded]

^{12.} Integrating NNMi with a Directory Service Through LDAP

12.6.1 nms-auth-config.xml

The nms-auth-config.xml file contains the settings for communicating with and building LDAP queries to the directory service in the XML format. This section provides a reference of only the elements that are relevant for LDAP configuration.

This file is located as follows:

- Windows: %NnmDataDir%nmsas\NNM\conf
- Linux: \$NnmDataDir/nmsas/NNM/conf

By default, the nms-auth-config.xml file available in this location does not contain the XML elements required for LDAP configuration.

You can manually add all the necessary XML elements to this file by following the instructions in this section.

NNMi places a sample nms-auth-config.xml file in a different location, which can be used for reference.

The sample nms-auth-config.xml file is available in the following location:

- Windows: %NnmInstallDir%newconfig\HPOvNnmAS\nmsas\conf
- Linux: \$NnmInstallDir/newconfig/HPOvNnmAS/nmsas/conf

О Тір

You can also copy the entire <ldapLogin> element from the sample nms-auth-config.xml file, and then make necessary modifications.

After editing the nms-auth-config.xml file (%NnmDataDir%nmsas\NNM\conf (Windows) or \$NnmDataDir/nmsas/NNM/conf (Linux)), force NNMi to read the LDAP configuration again by running the following command:

• Windows:

```
%NnmInstallDir%bin\nnmldap.ovpl -reload
```

• Linux:

\$NnmInstallDir/bin/nnmldap.ovpl -reload

```
rations are specified, to skip non-applicable LDAP servers to reduce log-on time. -->
    <filter>
      <usernamePattern>.*@hpe\.com</usernamePattern>
    </filter>
<!-- Time limit for performing searches against the LDAP server -->
    <searchTimeLimit>30000</searchTimeLimit>
    <connectTimeLimit>10000</connectTimeLimit>
<!-- Define at least one server URL -->
    <server>
      <hostname>ldaps://ldap.domain1.com</hostname>
      <secure>true</secure>
    </server>
<!---Optional. Bind credential and encrypted password for connecting to LDAP servers
that do not support anonymous access. Use "nnmldap.ovpl -encrypt" to create the encry
pted password.--->
    <bindCredential>
      <bindDN>someUser@some.com</bindDN>
      <bindCredential>someEncryptedPassword</bindCredential>
    </bindCredential>
<!-- This element defines the rules to search for users in this LDAP configuration --
>
    <users>
<!-- Optional. Filter that is matched against the user that attempts to log on. The i
ntention is to skip nonapplicable LDAP configurations to reduce the log-on time. Note
 that this is a Java regular expression.-->
      <filter>
        <usernamePattern>.*some\.com</usernamePattern>
      </filter>
<!-- Optional. The display name expression to show in the NNMi console.-->
      <displayName>${sn},${givenName} (HPE)</displayName>
<!-- Optional. Default roles that are given to all users that are authenticated again
st this configuration -->
      <defaultRoles>
        <role>guest</role>
      </defaultRoles>
<!-- One or more search configuration for locating user accounts. The pattern "{0}" i
n the string will be replaced with the log-on name entered by the user in the log-on
screen. -->
      <userSearch>
        <base>uid={0}</base>
        <baseContextDN>ou=People,o=domain.com</baseContextDN>
      </userSearch>
    </users>
<!-- Defines the rules to search for user roles or groups in this LDAP configuration
-->
    <roles>
<!-- Optional. Filter that defines which users should be attempted for role lookup ag
ainst this configuration. Note that this is a Java regular expression. -->
      <filter><usernamePattern>x</usernamePattern></filter>
<!-- One or more search configuration for locating LDAP groups that contain the authe
nticated user DN. Use the string "{1}" where the user's DN would appear. -->
      <roleSearch>
        <roleBase>member={1}</roleBase>
        <roleContextDN>ou=Groups,o=some.com</roleContextDN>
      </roleSearch>
      <roleSearch>
        <roleBase>GroupMember={1}</roleBase>
        <roleContextDN>CN=Groups,DC=mycompany,DC=com</roleContextDN>
      </roleSearch>
    </roles>
  </configuration>
</ldapLogin>
```

^{12.} Integrating NNMi with a Directory Service Through LDAP

12.7 Switching to the nms-auth-config.xml File

When the product is upgraded from 11-10 to 11-50 or later, the settings of the file ldap.properties are automatically migrated to the file nms-auth-config.xml.



Important

If the product is upgraded to 11-50 or later with defaultRole commented out in the file ldap.properties (as shown below), the file nms-auth-config.xml generated during the upgrade cannot be used as is.

#defaultRole=guest

Comment out or delete the following part, and then run the command nnmldap.ovpl -reload to reload the LDAP settings file.

<defaultRoles> <role/> </defaultRoles>

12. Integrating NNMi with a Directory Service Through LDAP

13

Managing Overlapping IP Addresses in a NAT Environment

NAT (network address translation) helps you save IP addresses by allowing many local networks to be connected to the global Internet using a single dynamic external (public) IP address. NAT can also enhance the security of private networks by hiding internal addresses from external networks. This chapter explains the configurations of NAT and overlapping IP addresses used by NNMi.

13.1 About NAT

Normally, network address translation is used to interconnect local networks with external networks. This technology has been developed as a solution to the need for more IPv4 addresses. Also, the demand for technologies such as NAT had been increasing because a specific range of IP addresses (see RFC 1918) had been designated for internal use only (routing on the Internet is not supported).

NAT translates IP header information. It translates the internal addresses of IP packets that need to pass through public networks into external addresses. NAT uses static and dynamic external addresses to translate internal addresses to external addresses.

^{13.} Managing Overlapping IP Addresses in a NAT Environment

13.2 Benefits of NAT

NAT provides the following benefits:

- IP address spaces can be saved because many hosts connect to the global Internet by using a single dynamic external IP address.
- Private IP addresses can be reused.
- The security of private networks can be enhanced by hiding the internal addresses from external networks.

^{13.} Managing Overlapping IP Addresses in a NAT Environment

13.3 Supported NAT types

NNMi supports the following NAT protocols:

• Static NAT

With static NAT, each internal IP address is always mapped to the same external IP address (each node has a pair of an internal address and an external address). This type of NAT allows accesses on the Internet while an unregistered (private) IP address is assigned to the internal host (such as a Web server).

• Dynamic NAT

Dynamic NAT is a scheme that enables the binding of external and internal addresses to be changed at each session. An internal IP address is mapped to a public IP address that is obtained from a pool of available registered (public) IP addresses. Normally, a table of registered IP addresses is maintained within the NAT router in the network. When an access from an internal IP address to the Internet is requested, the router selects from the table another internal IP address that is available.

• Dynamic port address translation (dynamic PAT) (also called a network address and port translation (NAPT)) Dynamic PAT is a type of NAT that translates not only IP addresses but also port numbers. If both addresses and port numbers are translated, multiple internal addresses can be used to communicate on the Internet concurrently using a single external address.

^{13.} Managing Overlapping IP Addresses in a NAT Environment

13.4 How to implement NAT in NNMi

NNMi manages a NAT environment by identifying each node that uses a tenant/IP address pair. NNMi administrators create a tenant definition for each NAT address domain. The tenant identifies a logical grouping of nodes. For example, an Internet provider's network might have multiple customers who implemented private IP addresses. Within NNMi, the Internet provider can assign each customer's nodes to a specific tenant name that identifies the customer. Within such a logical tenant grouping:

- NNMi administrators use discovery seeds to identify the tenant's member nodes on the basis of tenant/IP address pairs.
- Subnet connection rules apply independently within each tenant's group of nodes.
- Router redundancy groups are monitored within each tenant, independently from any other tenant's group of nodes.
- NNMi discovers L2 Connections only within each tenant's group of nodes, and between that defined tenant's nodes and nodes assigned to a tenant named default tenant.
- Assign any infrastructure device that interconnects multiple NAT domains (such as the NAT gateway router) to the default tenant. This ensures that NNMi displays the layer 2 connections your work group (and customers) needs to see.
- Security groups determine how many tenants an NNMi user can see. An assigned security group can include nodes from more than one tenant. For details, see 14. NNMi Security and Multi-Tenancy.



A best practice is to have no duplicate domain name system (DNS) names across all NAT domains in your network management environment.

The NNMi implementation method and requirements vary depending on the NAT protocol you are using. For example, use of dynamic NAT or PAT requires additional hardware and licenses. See the appropriate section listed in the following based on your NAT protocol:

- 13.5 Considerations on static NAT
- 13.6 Dynamic NAT and dynamic PAT considerations

For details, see 13.6.6 Deploying NNMi in a network address translation (NAT) environment and 13.6.7 NNMi calculations for state and status.

13. Managing Overlapping IP Addresses in a NAT Environment

If each instance is configured as a unique tenant, one NNMi management server can monitor any number of static NAT instances. For details about tenants, see 14. NNMi Security and Multi-Tenancy and *Configure Tenants* in NNMi Help.

The following figure shows an example of a static NAT configuration.





Legend:

External: External address Internal: Internal address

The node belonging to the default tenant can establish a Layer-2 connection with any of the nodes in any of the tenants. A node in a non-default tenant can establish a Layer 2 connection only with the devices in the same tenant or the default tenant.

Subnets are specific to tenants (a subnet does not span multiple tenants). The benefit of subnets is that the same subnet can be used by different tenants.

A router redundancy group (RRG) cannot span multiple tenants.

^{13.} Managing Overlapping IP Addresses in a NAT Environment

Allocate all infrastructure devices that interconnect to multiple NAT domains (such as NAT gateways) in the default tenant. This ensures that the Layer 2 connections that work groups (and clients) must check are displayed in NNMi.

The devices in the default security group are displayed in all views. To control access to a device, assign that device to a security group that is not the default security group.

13.5.1 Hardware and software requirements for static NAT

There are no special hardware or software requirements for static NAT. One NNMi management server can manage any number of static NAT domains with either NNMi or NNMi Advanced.

13.5.2 Communication using static NAT

NNMi communicates successfully through a static NAT firewall by automatically applying any available overlapping address mappings to determine the tenant/external IP address pairs for static NAT communications. For details about the benefits, see 13.7 Mapping overlapping IP addresses.

(1) Administering ICMP polling of the management addresses in a static NAT environment

In a NAT environment, a firewall blocks NNMi from communicating with NAT nodes using the IP addresses on the nodes (the private IP addresses). To remedy this, the NAT address (the public IP address) is used for communication with NNMi.

In a NAT environment, a node's management address might be different from the IP addresses hosted on the node. For NNMi to discover a node in a NAT environment, you must add the NAT address to NNMi as a discovery seed. NNMi uses this NAT address for communication, even though it is not in the node's <code>ipAddressTable</code>.

By providing this feature, NNMi avoids false node down incidents and offers a better root cause analysis.

(2) Overview of ICMP polling of the management addresses in a NAT environment

(a) ICMP polling of the management addresses in a NAT environment

If you have a NAT environment, we recommend that you do not disable this setting.

To enable ICMP management address polling (if it has been disabled):

- 1. From the workspace navigation panel, select the **Configuration** workspace, expand the Monitoring folder, select **Monitoring Configuration**, and locate the **Default Settings** tab.
- 2. In the **ICMP Fault Monitoring** section, select **Enable Management Address Polling**. See *Set Default Monitoring* in NNMi Help.

View the information NNMi displays after performing Actions -> Monitoring Settings for SNMP agents. The displayed information indicates whether management address polling is enabled for NNMi.

When ICMP management address polling is enabled, NNMi changes as follows:

• The Management Address ICMP State field appears in the following forms and table views:

^{13.} Managing Overlapping IP Addresses in a NAT Environment

- Node form
- SNMP Agent form
- **SNMP Agent** table views
- NNMi changes the display location of the management address ICMP state, as well as the way it determines the SNMP agent status.

The following table shows the management address ICMP and IP address state polling actions that NNMi takes for ICMP management address polling and ICMP fault polling settings.

Table 13-1: ICMP configurations and	resulting state	polling
-------------------------------------	-----------------	---------

ICMP management address polling	ICMP fault polling	Management ICMP address state	IP address state
Enabled [#]	Disabled [#]	Polled [#]	Not polled [#]
Enabled	Enabled	Polled	Polled
Disabled	Disabled	Not polled	Not polled
Disabled	Enabled	Not polled	Polled

#: Default setting

The table below shows the SNMP agent statuses and the variation in generated incidents, which are determined by APA based on the responses from the SNMP agent and the management address ICMP. With ICMP polling of management addresses enabled, APA considers the management address ICMP response and the SNMP agent response when generating conclusions and incidents.

Table 13-	-2: Determini	ng SNMF	Pagent statu	s and ge	nerated incic	lents

SNMP agent response	Management address ICMP response	SNMP agent status	Generated incident
Responding	Responding	Normal	None
Responding	Not responding	Minor	The following incidents might be generated by other network problems:NoneAddressNotResponding
Not responding	Responding	Critical	SNMPAgentNotResponding
Not responding	Not responding	Critical	The following incidents might be generated by other network problems:NoneNodeDown

13.5.3 Discovery and static NAT

The NNMi administrator must create a tenant definition to identify each static NAT domain within your network management environment. Spiral discovery requires discovery seeds (tenant and address pairs) to identify each node before NNMi can discover and monitor each node.

The NNMi administrator must create a discovery seed for each node in the static NAT domain. A discovery seed must provide the following information for each node:

^{13.} Managing Overlapping IP Addresses in a NAT Environment
- External IP address (public address from the external/internal IP address pair)
- Tenant name

For details, see NNMi Help.

When you add discovery seeds to a static NAT environment (by using the nnmloadseeds.ovpl command or the NNMi console), make sure that you use a node's external (public) IP address. For details, see the *nnmloadseeds.ovpl Reference Page*.

We recommend that you use a domain name system (DNS) name that is not duplicated.

13.5.4 Monitoring configuration for static NAT

Depending on your network environment, the NNMi administrator can select the ICMP fault monitoring settings to be used. See also 13.6.7 NNMi calculations for state and status.

• Monitoring Configuration > Node Settings tab

Configure monitoring for a node group. In this case, make your choices in the **ICMP Fault Monitoring** section. For details, see NNMi online Help.

- Management address polling (enabled by default and highly recommended)
- IP address fault polling (optional)
- Monitoring Configuration > Default Settings tab

In this case, make your choices in the ICMP Fault Monitoring section. For details, see NNMi online Help.

Important

If your network environment also includes any dynamic NAT domains, default settings might not be appropriate because you might want different settings for the static NAT domains from those for the dynamic NAT domains.

13.5.5 Traps and static NAT

For an NNMi management server to be able to receive SNMP traps from the nodes located behind a NAT gateway, you must change the managed nodes. This subsection explains two types of SNMP traps, SNMPv2c and SNMPv1.

NNMi must unambiguously resolve the source address of each trap that it receives.

(1) SNMPv2c traps

The figure below shows the format of an SNMPv2c trap. In this figure, the top section forms the IP header and the lower section forms the SNMP Trap Protocol Data Unit (PDU).

^{13.} Managing Overlapping IP Addresses in a NAT Environment

SNMPv2c Trap Format
Version and other information
Source Address
Destination Address
PDU-Type: 7
Request Identifier
Error Status
Error Index
PDU Variable Bindings

SNMPv2c traps do not include an Agent Address field in the PDU section. Therefore, the trap's only source field is located in the IP packet header. The source field is translated by the NAT router appropriately.

Make sure that the sources of all traps of devices located behind the NAT router are recognized by the interface associated with the source node's private internal IP address. This ensures that the NAT gateway can translate a trap to the appropriate public address.

The figure below shows an example of correct translation from a NAT gateway. The NAT gateway properly translates a trap that begins with the source address of 192.168.1.2 to address 15.2.12.2. Then the NNMi management server correctly resolves this address.

^{13.} Managing Overlapping IP Addresses in a NAT Environment



External: External address

Internal: Internal address

(2) SNMPv1 traps

An SNMPv1 trap includes an Agent Address field in the PDU. The figure below shows the format of SNMPv1 traps. In this figure, the IP header forms the top section and the SNMP trap PDU forms the lower section.

^{13.} Managing Overlapping IP Addresses in a NAT Environment

SNMPv1 Trap Format
Version and other information
Source Address
Destination Address
PDU-Type: 4
Enterprise
Agent Address
Generic Trap Code
Specific Trap Code
Timestamp
PDU Variable Bindings

Because the Agent Address field is embedded in the PDU and not in the header, the NAT router does not normally translate this value. To configure NNMi to recognize the header address and ignore the payload's agent address:

- 1. Edit the following file:
 - Windows: %NNM_PROPS%\nms-jboss.properties
 - Linux: \$NNM PROPS/nms-jboss.properties
- 2. Locate the following line:

#!com.hp.nnm.trapd.useUdpHeaderIpAddress=false

3. Change the value to true and remove the characters #!, as shown below:

com.hp.nnm.trapd.useUdpHeaderIpAddress=true

- 4. Save the changes.
- 5. Execute the following commands to restart NNMi:

```
ovstop
ovstart
```

The following figure shows an example of an SNMPv1 trap where NNMi ignores the conflicting Agent Address fields.

13. Managing Overlapping IP Addresses in a NAT Environment

Figure 13-3: SNMPv1 example



gena: External: External address Internal: Internal address

NNMi provides the following Custom Incident Attributes (CIAs):

^{13.} Managing Overlapping IP Addresses in a NAT Environment

- cia.agentAddress: This is the IP address stored in the SNMPv1 trap data of the SNMP agent that generated the trap.
- cia.internalAddress: If the network management domain supports static NAT, the NNMi administrator can configure this attribute to display the internal IP address that is mapped to the external management address of the selected incident's source node.

You must use the **Overlapping IP Address Mapping** form to map the external management IP address (public address) to this internal address (private address). For details, see NNMi Help.

13.5.6 Subnets and static NAT

Note the following subnet and NAT considerations:

- Subnets are unique to tenants (a subnet does not span multiple tenants). The benefit of subnets is that the same subnet can be used by different tenants.
- Subnet filters use tenant and address pairs.
- If subnet connection rules are configured, they apply to all tenants. The subnet members must be unique in all tenants (a node is assigned to only one tenant). You can use subnet connection rules to establish a link between a tenant and the default tenant. This feature is supported only when one of the two linked tenants is the default tenant.

13.5.7 Global network management and static NAT

At least one static or routable (non-translated) address must exist per regional manager. This enables NNMi management servers to communicate with each other, keeping communications internal and secure. For details about global network management, see 15. Global Network Management.

^{13.} Managing Overlapping IP Addresses in a NAT Environment

13.6 Dynamic NAT and dynamic PAT considerations

One NNMi management server can manage one dynamic NAT or dynamic PAT domain. All nodes in such a domain must belong to the same unique tenant. The NNMi management server must participate in a global network management environment as a regional manager. For details, see the example of a dynamic NAT configuration shown below.

If a regional manager is located behind a NAT firewall, its external (public) address must be static.





To monitor multiple dynamic NAT and dynamic PAT domains, use NNMi's global network management feature. Tenants must be unique within the entire NNMi global network management configuration. For details, see the following figure for an example of a global network management configuration within a NAT environment.

^{13.} Managing Overlapping IP Addresses in a NAT Environment





Legend:

External: External address Internal: Internal address

Devices that belong to the default tenant can have a Layer 2 connection to any device in any tenant. Devices within any tenant other than the default tenant can have a Layer 2 connection only to devices within the same tenant or the default tenant.

^{13.} Managing Overlapping IP Addresses in a NAT Environment

Assign all infrastructure devices that interconnect multiple NAT domains (such as the NAT gateway) to the default tenant. This ensures that NNMi displays the Layer 2 connections your workgroup (and customers) needs to see.

The devices in the default security group are displayed in all views. To control access to a device, assign that device to a security group that is not the default security group.

For details about global network management, see 15. Global Network Management. For details about configuring tenants, see *Configure Tenants* in NNMi Help.

13.6.1 Hardware and software requirements for dynamic NAT and dynamic PAT

NNMi Advanced software is required for dynamic NAT and dynamic PAT environments. Each dynamic NAT or dynamic PAT domain requires its own NNMi regional manager.

13.6.2 Discovery and dynamic NAT or dynamic PAT

NNMi uses tenants to support networks containing overlapping address domains. Overlapping address domains might exist in the dynamic NAT or dynamic PAT area in the network management domain. To handle such a network, NNMi places overlapping address domains in different tenants (by using seeded discovery). For details, see NNMi Help.

When adding discovery seeds (with the nnmloadseeds.ovpl command or from the NNMi console) in a dynamic NAT or PAT environment, make sure that you use the internal IP addresses of the nodes.

For details, see the nnmloadseeds.ovpl Reference Page or NNMi Help.

13.6.3 Monitoring configuration for dynamic NAT

Depending on your network environment, the NNMi administrator can choose to use the ICMP fault monitoring settings. See also 13.6.7 NNMi calculations for state and status.

• Monitoring Configuration > Node Settings tab

Configure monitoring for a node group. In this case, make your choices in the **ICMP Fault Monitoring** section. For details, see NNMi online Help.

- Management address polling (enabled by default and highly recommended)
- IP address fault polling (optional)
- Monitoring Configuration > Default Settings tab

In this case, make your choices in the ICMP Fault Monitoring section. For details, see NNMi online Help.

Important

If your network environment also includes any static NAT domains, default settings might not be appropriate because you might want different settings for static NAT domains from those for dynamic NAT domains.

13. Managing Overlapping IP Addresses in a NAT Environment

13.6.4 Subnets and dynamic NAT or dynamic PAT

Note the following subnet and dynamic NAT and dynamic PAT considerations:

- Subnets are unique to tenants (a subnet does not span multiple tenants). The benefit of subnets is that the same subnet can be used by different tenants.
- Subnet filters use tenant and address pairs.
- If subnet connection rules are configured, they apply to all tenants. The subnet members must be unique in all tenants (a node is assigned to only one tenant). You can use subnet connection rules to establish a link between a tenant and the default tenant. This feature is supported only when one of the two linked tenants is the default tenant.

13.6.5 Global network management and dynamic NAT or dynamic PAT

At least one static or routable (non-translated) address must exist per regional manager. This enables NNMi management servers to communicate with each other, keeping communications internal and secure.

If a regional manager is located behind a NAT firewall, its external address must be static.

For details about global network management, see 15. Global Network Management. See also *Tenant Best Practices for Global Network Management* in NNMi Help.

13.6.6 Deploying NNMi in a network address translation (NAT) environment

To deploy NNMi in a NAT environment:

- 1. Identify and compile a list of the NAT domains in your network management environment.
- 2. Determine the type of supported NAT that is used within each NAT domain.
- 3. Deploy each NNMi management server as required in relation to each NAT domain (inside or outside the NAT domain's internal IP address space).

See the following special considerations:

- 13.5 Considerations on static NAT
- 13.6 Dynamic NAT and dynamic PAT considerations
- 4. Use the NNMi **Configuration** > **Discovery** > **Tenants** workspace to define a unique tenant name for each NAT domain.

Important

If you are using global network management in your deployment, this name must be unique among all NNMi management servers (regional managers and the global manager).

- 5. Determine the nodes within each NAT domain that NNMi needs to monitor.
- 6. Only for static NAT domains: Create any overlapping address mappings to identify each node's assigned NAT external/internal IP address pair.

For the benefits of creating overlapping address mappings, see 13.7 Mapping overlapping IP addresses. Provide the following information:

^{13.} Managing Overlapping IP Addresses in a NAT Environment

- Tenant name
- External IP address
- Internal IP address

Use either the NNMi Configuration > Discovery > Overlapping Address Mappings workspace or the nnmloadipmappings.ovpl command line tool.

For details, see NNMi online Help.

- 7. Depending on where the NNMi management server is deployed in your network environment, a firewall might block NNMi from communicating with nodes in a NAT domain when NNMi uses the node's internal addresses. Therefore, for the Configuration > Communication Configuration setting, use the appropriate Preferred Management Address setting (NAT's external or internal IP address).
- 8. Verify the **Monitoring Configuration** settings for NAT in your network environment:
 - 13.5.4 Monitoring configuration for static NAT
 - 13.6.3 Monitoring configuration for dynamic NAT

For details about Monitoring Configuration, see NNMi online Help.

9. Configure a discovery seed for each node.

Important

Assign any infrastructure device that interconnects multiple NAT domains (such as a NAT gateway router) to the default tenant.

Use either the NNMi Configuration > Discovery > Seeds workspace or the loadseeds.ovpl command line tool:

- If the NNMi management server is inside the internal IP address space, configure discovery seeds using the internal IP address:
 - Host name/IP (use the internal IP address)
 - Tenant name
- If the NNMi management server is outside the internal IP address space, configure discovery seeds using the external IP address:.
 - Host name/IP (use the external IP address)
 - Tenant name

For details, see NNMi online Help.

10. Verify that NNMi discovery found the nodes you expected.

If not, double-check your configurations (above).

- 11. Verify that the NNMi settings meet your team's needs:
 - Fine-tune the security group assignment of each node to manage which team members/customers can see each node at the NNMi console. Use NNMi Configuration > Security > Security Groups workspace.
 - Review the monitoring configuration settings that apply to these nodes and fine-tune as necessary. Use the NNMi Configuration > Monitoring > Monitoring Configuration workspace.
- 12. Verify that the connections between nodes appear on the NNMi maps as expected.

If not, do the following:

- Verify that both nodes involved in a connection have proper tenant assignments (default tenant or other tenant).
- Verify that the Subnet Connection Rules tab settings in Configuration > Discovery > Discovery Configuration are correct.

^{13.} Managing Overlapping IP Addresses in a NAT Environment

- To force NNMi to add connections that are not found automatically, use the nnmconnedit.ovpl command line tool. For details, see the reference page for the nnmconnedit.ovpl command.
- 13. Review the SNMP trap forwarding rules configured in each node's SNMP agent to include the appropriate NNMi management server's IP address.
- 14. Only for static NAT domains: Configure the SNMP agent on each static NAT node to ensure that the interface associated with **Internal Address** in NNMi's **Overlapping Address Mappings** sources all traps that are sent to the NNMi management server.
- 15. If your network environment includes SNMPv1, make the appropriate required changes to the NNMi configuration. See 13.5.5 Traps and static NAT.

13.6.7 NNMi calculations for state and status

By default, NNMi automatically enables ICMP polling of each node's management address, including nodes residing in a NAT environment (**Configuration** > **Monitoring** > **Monitoring Configuration**, the **Default Settings** tab, **ICMP Fault Monitoring** section's **Enable Management Address Polling** setting). If you have a NAT environment, we recommend that you do not disable this setting.

Important

In the **Inventory** > **Node** view, select a node and use the **Actions** > **Configuration Details** > **Monitoring Settings** command. The displayed information indicates whether NNMi has this management address polling enabled

When management address polling is enabled, the Agent ICMP State field appears in the following locations:

- Node form
- SNMP Agent form
- SNMP Agent table views

The following table shows how NNMi behavior changes based on the ICMP Fault Monitoring settings.

ICMP fault monitoring setting	IS	Management ICMP address s IP address state	tate
Enable management address polling	Enable IP address fault polling	Agent ICMP state	IP address state
Enabled [#]	Disabled [#]	Polled	Not polled
Enabled	Enabled	Polled	Polled
Disabled	Disabled	Not polled	Not polled
Disabled	Enabled	Not polled	Polled

#: Default settings

When management address polling is enabled, NNMi considers both the management address's ICMP response and the SNMP agent's response when calculating conclusions and generating incidents.

^{13.} Managing Overlapping IP Addresses in a NAT Environment

The following table shows the SNMP agent status calculations determined by the combined ICMP and SNMP responses.

Table 13-4: Determining SNMP agent status

SNMP agent's response	Management address's ICMP response	Resulting SNMP agent status
Responding	Responding	Normal
Responding	Not responding	Minor
Not responding	Responding	Critical
Not responding	Not responding	Critical

^{13.} Managing Overlapping IP Addresses in a NAT Environment

13.7 Mapping overlapping IP addresses

If a network management environment contains overlapping address domains, each such domain must be configured as a unique tenant. For details, see *Configure Tenants* in NNMi Help and 14. NNMi Security and Multi-Tenancy.

If your network management domain supports static NAT but the NNMi management server is not within the scope of static NAT, you can configure NNMi to display the NAT external IP address (public address) in the **Mapped Address** attribute in the **IP Address** form for the identified tenant/NAT internal IP address (such as a private IPv4 address) pairs.

If you are configuring NNMi for areas of your network management domain that use dynamic NAT or dynamic PAT, do not use the **Overlapping IP Address Mapping** form. For details, see 13.6 Dynamic NAT and dynamic PAT considerations.

A static NAT configuration for a network domain might be applied to public IP addresses or private IP addresses, or both.

To configure NNMi to display the static NAT external IP address in the **Mapped Address** attribute in the **IP Address** form for the identified tenant/NAT internal IP address pairs, do one of the following:

- On the NNMi console, use the Overlapping IP Address Mapping form.
- Use the nnmloadipmappings.ovpl command.

For details, see NNMi Help or the nnmloadipmappings.ovpl Reference Page.

13.7.1 Ranges of private IP addresses

The Internet Engineering Task Force (IETF) and Internet Assigned Numbers Authority (IANA) reserved the following IP address ranges for private networks, such as enterprise local area networks (LANs), corporate offices, and residential networks:

IPv4 private address ranges (RFC 1918):

- 10.0.0.0 to 10.255.255.255 (24-bit block)
- 172.16.0.0 to 172.31.255.255 (20-bit block)
- 192.168.0.0 to 192.168.255.255 (16-bit block)

IPv6 private address ranges:

- fc00::/7 address block = RFC 4193 unique local addresses (ULA)
- fec0::/10 address block = deprecated (RFC 3879)

13. Managing Overlapping IP Addresses in a NAT Environment

14

NNMi Security and Multi-Tenancy

In NNMi, security and multi-tenancy provide for restricting user access to information about the objects in the NNMi database. This restriction is useful for customizing the views of network operators to their areas of responsibility. It also supports service providers with per-organization configuration of NNMi. This chapter describes the NNMi security and tenant models and gives suggestions for configuration. By default, all NNMi console users can see information for all objects in the NNMi database. If this default configuration is acceptable for your environment, you do not need to read this chapter.

14.1 Effects of limiting object access

Configuring NNMi security has the following impacts:

Topology inventory objects:

- Each NNMi console user sees only those nodes that match the configuration for that user's NNMi user account.
- Sub-node objects, such as interfaces, inherit the access control from the node.
- Inter-node objects, such as connections, are visible only if the NNMi console user can see at least one of the nodes involved.
- An NNMi console user sees only those node groups for which that user can access at least one node in the group.

Maps and path views:

- Maps show connections for which the NNMi console user has permission to view both of the participating nodes.
- Path views omit or show as clouds any intermediate nodes to which the NNMi console user does not have access.

Incidents:

- For incidents whose source node is in the NNMi topology, an NNMi console user sees only the incidents for which the user has access to the source node.
- Incidents that do not have a source node, such as NNMi health and licensing management event incidents, are handled as a group. The NNMi administrator determines which NNMi console users see them (by associating the users with the **Unresolved Incidents** security group).
- Incidents that result from traps for which the source node is not in the NNMi topology are handled in the same way as incidents with no source node. If NNMi is configured to generate these incidents, the NNMi administrator determines which NNMi console users see them (by associating the users with the **Unresolved Incidents** security group).

The incident assignment action does not check user access. It is possible for an NNMi administrator to assign an incident to an NNMi console user who does not have permission to view that incident.

NNMi console actions:

- For actions that run without any selections, an NNMi console user sees only those actions the user has permission to run.
- For actions that run against one or more selected objects, an NNMi console user must have the correct access level to the selected objects. Depending on the security configuration, the NNMi console might present actions that are not valid on some of the objects visible in the NNMi console views. Invoking one of these actions results in an error message regarding this limitation.
- For map views, NNMi cannot distinguish between unknown nodes and nodes that exist in the NNMi topology but are not accessible by the current user.

MIB browser and Line Grapher:

- An NNMi console user can view MIB data and graphs for nodes to which the user has access.
- An NNMi console user can view MIB data for nodes for which the user knows the SNMP community string.

NNMi console URLs:

Users must sign in to NNMi before accessing an NNMi console view from a direct URL. NNMi enforces that user's access according to the NNMi security configuration and limits the available topology accordingly.

14.2 The NNMi security model

The NNMi security model provides user access control to the objects in the NNMi database. This model is appropriate for use by any network management organization that wants to limit NNMi user access to specific objects and incidents. The NNMi security model has the following benefits:

- Provides a way to limit an NNMi console operator's view of the network. Operators can focus on specific device types or network areas.
- Provides for customizing operator access to the NNMi topology. The level of operator access can be configured per node.
- Provides for filtering the Nodes (All Attributes) view by security group.
- Simplifies the configuration and maintenance of node groups that align with the security configuration.
- Can be used independently of the NNMi tenant model.

Possible use cases for NNMi security include the following:

- Provide NNMi operator focus on equipment type within a site (custom maps).
- Provide NNMi operators at different sites views that show only the nodes at a given site (custom maps).
- Stage nodes during deployment. NNMi administrators see all nodes, while NNMi operators see only the deployed nodes.
- Provide full access to all NOC operators, and limit access to NOC customers.
- Provide full network views to the central NOC operators, and limit the views of the regional NOC operators.

14.2.1 Security groups

In the NNMi security model, user access to nodes is controlled indirectly though user groups and security groups. Each node in the NNMi topology is associated with only one security group. A security group can be associated with multiple user groups.

Each user account is mapped to the following user groups:

One or more of the following preconfigured NNMi user groups:

- NNMi Administrators
- NNMi Global Operators
- NNMi Operator Level 1
- NNMi Operator Level 2
- NNMi Guest Users

This mapping is required for NNMi console access and determines the actions that are available within the NNMi console. If a user account is mapped to more than one of these NNMi user groups, the user receives the superset of the permitted actions.

The NNMi Web Services Clients user group does not grant access to the NNMi console; however, it does grant administrator-level access to all NNMi objects.

The NNMi Global Operators user group (globalops) grants access to topology objects only. A user must be assigned to one of the other user groups (level2, level1, or guest) to access the NNMi console.

The administrator must not map the globalops user group to any security group because this user group is, by default, mapped to all security groups.

Custom user groups that are mapped to security groups.

These mappings provide access to objects in the NNMi database. Each mapping includes an object access privilege level that applies to the nodes for a security group. The object access privilege level also applies to the related database objects, such as interfaces and incidents. For example, a user with Object Operator Level 1 access to node A containing interfaces X and Y has Object Operator Level 1 access to all of the following database objects:

- Node A
- Interfaces X and Y
- Incidents whose source object is node A, interface X, or interface Y

NNMi provides the following security groups:

Default Security Group

In a new NNMi installation, the initial security group assignment for all nodes is the **Default Security Group**. This means that the default is that all users can see all objects in the **Default Security Group**. The NNMi administrator can configure the nodes that are to be associated with the **Default Security Group** and the users who can access the objects in the **Default Security Group**.

Unresolved Incidents

The **Unresolved Incidents** security group provides access to incidents that NNMi creates from received traps whose source node is not in the NNMi topology. By default, all users can see all incidents associated with the **Unresolved Incidents** security group. The NNMi administrator can configure the users who are permitted to access the incidents associated with the **Unresolved Incidents** security group.

All sensors inherit the node's security group assignment.

Best practices

The following are best practices for configuring NNMi security:

- Map each user account to only one preconfigured NNMi user group.
- Do not map the preconfigured NNMi user groups to security groups.
- Because any user account mapped to the **NNMi Administrators** user group receives administrator-level access to all objects in the NNMi database, do not map this user account to any other user groups.
- Create a separate user account for the Web Services Client role. Because this user account has access to the entire NNMi topology, map this user account to only the **NNMi Web Service Clients** user group.

14.2.2 Example security group structure

The three user frames in the figure below indicate for this example NNMi topology the primary groupings for which users need to view the nodes. For complete user access control, each of the four unique subgroups corresponds to a unique security group. Each unique security group can be mapped to one or more user groups to represent the available levels of user access to the objects in that security group.

Table 14-1 lists the mappings between the security groups and the possible custom user groups for this topology. An actual implementation of this security model might not require all of these custom user groups.

Table 14-2 lists the mappings for several user accounts and the user groups for this topology.

^{14.} NNMi Security and Multi-Tenancy



Figure 14-1: Example topology for user access requirements

Table 14-1: Example security group mappings

Security group	Security group's nodes	User groups	Object access privileges
SG1	A, B, C	UG1 Administrator	Object Administrator
		UG1 Level 2	Object Operator Level 2
		UG1 Level 1	Object Operator Level 1
		UG1 Guest	Object Guest
SG2	D, E	UG2 Administrator	Object Administrator
		UG2 Level 2	Object Operator Level 2
		UG2 Level 1	Object Operator Level 1
		UG2 Guest	Object Guest
SG3	F, G	UG3 Administrator	Object Administrator
		UG3 Level 2	Object Operator Level 2
		UG3 Level 1	Object Operator Level 1
		UG3 Guest	Object Guest
SG4	H, I, J	UG4 Administrator	Object Administrator
		UG4 Level 2	Object Operator Level 2
		UG4 Level 1	Object Operator Level 1
		UG4 Guest	Object Guest

Table 14-2: Example user account mappings

User account	User groups	Node access	Notes
User Q	NNMi Level 2 Operators	None	This user has operator level 2 access to the
	UG1 Level 2	A, B, C	nodes in the user Q frame.
	UG2 Level 2	D, E	

User account	User groups	Node access	Notes
User Q	UG3 Level 2	F,G	This user has operator level 2 access to the nodes in the user Q frame.
User R	NNMi Level 1 Operators	None	This user has operator level 1 access to the
	UG2 Level 1	D, E	nodes in the user R frame.
User S	NNMi Level 2 Operators	None	This user has operator level 2 access to the
	UG3 Level 2	F, G	nodes in the user S frame.
	UG4 Level 2	Н, I, J	
User T	NNMi Level 2 Operators	None	User T has access (with varying privilege
	UG1 Guest	A, B, C	topology.
	UG2 Administrator	D, E	This user has administrative access to nodes D and E but cannot see the menu
	UG3 Level 2	F, G	items for tools that require administrative
	UG4 Level 1	Н, I, J	management server, this user can run command-line tools that require administrative access against nodes D ar E only.

^{14.} NNMi Security and Multi-Tenancy

14.3 The NNMi tenant model

The NNMi tenant model provides strict segregation of topology discovery and data into tenants, also called organizations or customers. This model is appropriate for use by service providers, especially managed service providers, and large enterprises. The NNMi tenant model has the following benefits:

- Marks the organization to which each node belongs.
- Provides for filtering the Nodes (All Attributes) inventory view by tenant and security group.
- Meets regulatory requirements for separating operator access to customer data.
- Simplifies the configuration and maintenance of node groups that align with the tenant configuration.
- Simplifies configuration of NNMi security.

Use NNMi multi-tenancy to provide different customer views for a service provider that has multiple customers (tenants) managed from the same NNMi management server.

14.3.1 Tenants

The NNMi tenant model adds the idea of an organization to the security configuration. Each node in the NNMi topology belongs to only one tenant. The tenant provides logical separation in the NNMi database. Object access is managed through security groups.

For each node, the initial discovery tenant assignment occurs when the node is first discovered and added to the NNMi database. For seeded nodes, you can specify the tenant to assign to each node. NNMi assigns to the Default Tenant all other discovered nodes (those included in an auto-discovery rule but not seeded directly). An NNMi administrator can change the tenant for a node at any time after discovery.

Each tenant definition includes an initial discovery security group. NNMi assigns this initial discovery security group to the node along with the initial discovery tenant. An NNMi administrator can change the security group for a node at any time after discovery.

Changing the tenant assignment of a node does not automatically change the security group assignment.

NNMi provides the Default Tenant. This means that the default is that all NNMi users have access (through the **Default Security Group**) to all objects associated with this tenant.

All sensors inherit the tenant and security group assignments of the node.

Best practices

The following best practices apply to NNMi tenant configuration:

- For a small organization, a single security group per tenant is probably sufficient.
- You might want to subdivide a large organization into multiple security groups.
- To prevent users from accessing nodes across organizations, ensure that each security group includes nodes for only one tenant.

14.3.2 Example tenant structure

The figure below shows an example NNMi topology consisting of two tenants. The three frames for users L, M, and N indicate the primary groupings for which users need to view the nodes. The topology for Tenant 1 is managed as a single group, so it needs only one security group. The topology for Tenant 2 is managed in overlapping sets, so it is separated into three security groups.

Table 14-3 lists the mappings between the security groups and the possible custom user groups for this topology. (An actual implementation of this security model might not require all of these custom user groups.)

Table 14-4 lists the mappings for several user accounts and the user groups for this topology.



Figure 14-2: Example topology for multiple tenants

Table 14-3	Example security	aroun mannings	for multi	ole tenants
	Litample security	group mappings		

Security group	Security group's nodes	User groups	Object access privileges
T1 SG	A, B, C, D, E	T1 Administrator	Object Administrator
		T1 Level 2	Object Operator Level 2
		T1 Level 1	Object Operator Level 1
		T1 Guest	Object Guest
T2 SGa	F, G	T2_a Administrator	Object Administrator
	T2_a L	T2_a Level 2	Object Operator Level 2
		T2_a Level 1	Object Operator Level 1
		T2_a Guest	Object Guest
T2 SGb	Н	T2_b Administrator	Object Administrator
		T2_b Level 2	Object Operator Level 2
		T2_b Level 1	Object Operator Level 1
		T2_b Guest	Object Guest

Security group	Security group's nodes	User groups	Object access privileges
T2 SGC	I,J	T2_c Administrator	Object Administrator
		T2_c Level 2	Object Operator Level 2
		T2_c Level 1	Object Operator Level 1
		T2_c Guest	Object Guest

Table 14-4: Example user account mappings for multiple tenants

User account	User groups	Node access	Notes
User L	NNMi Level 2 Operators	None	This user has operator level 2 access to the
	T1 Level 2	A, B, C, D, E	all nodes in Tenant 1.
User M	NNMi Level 1 Operators	None	This user has operator level 1 access to the
	T2_a Level 1	F, G	subset of the nodes in Tenant 2.
	T2_b Level 1	Н	
User N	NNMi Level 2 Operators	None	This user has operator level 2 access to the
	T2_b Level 2	Н	subset of the nodes in Tenant 2.
	T2_c Level 2	I,J	

^{14.} NNMi Security and Multi-Tenancy

14.4 NNMi security and multi-tenancy configuration

NNMi security and multi-tenancy configuration applies to the entire NNMi database. Any NNMi administrator can view and configure operator access to all objects for all tenants.

After an NNMi administrator has defined at least one custom security group, the **Security Groups** field is visible on all **Node** forms and as a column in the **Nodes** and **Nodes** (All Attributes) inventory views.

After an NNMi administrator has defined at least one custom tenant, the **Tenant** field is visible on all **Node** forms and as a column in the **Nodes** and **Nodes** (All Attributes) inventory views.

Node groups

To create a node group that aligns with part of the security or multi-tenancy configuration, specify a node group additional filter based on security group UUID, security group name, tenant UUID, or tenant name. Use these node groups to configure per-security group or per-tenant polling cycles for monitoring and incident lifecycle transition actions.

🛛 Тір

Because security group and tenant names can change, specify the security group or tenant UUID in additional filters. This information is available on the configuration forms and in the nnmsecurity.ovpl command output.

User groups: NNMi console access

User account mapping to one of the predefined NNMi user groups sets the NNMi role and the visibility of menu items in the NNMi console. We recommend that each user account be granted the NNMi role that matches the highest object access privilege for that user's topology objects.

The exception to this recommendation is at the administration level because NNMi administrators can access all topology objects. To configure an NNMi console user as an administrator of only some nodes in the NNMi topology, assign that user to the NNMi Level 2 Operators or NNMi Level 1 Operators user group. Also assign that user to a custom user group mapped with the Object Administrator object access privilege to a security group containing a subset of the nodes in the topology.

User groups: directory service

If you are storing user group membership in the NNMi database, all object access configuration occurs in the NNMi configuration areas through user groups, user account mappings, security groups, and security group mappings.

If you are storing user group membership in a directory service, object access configuration is shared between NNMi configuration (security groups and security group mappings) and the directory service content (user group membership). Do not create user accounts or user account mappings in the NNMi database. For each applicable group in the directory service, create one or more user groups in the NNMi database. In NNMi, set the **Directory Service Name** field of each user group definition to the distinguished name of that group in the directory service.

For details, see 12. Integrating NNMi with a Directory Service Through LDAP.

14.4.1 Security and multi-tenancy configuration tools

NNMi provides several tools for configuring multi-tenancy and security.

Security Wizard

The **Security Wizard** in the NNMi console is useful for visualizing the security configuration. It is the easiest way to assign nodes to security groups within the NNMi console. The **View Summary of Changes** tab presents a list of unsaved changes from the current wizard session. It also identifies potential problems with the security configuration.

For details about using the Security Wizard, click NNMi Help links within the wizard.

Note

The Security Wizard is for NNMi security configuration only. It does not include tenant information.

NNMi console forms

The forms for individual security and multi-tenancy objects in the NNMi console are useful for concentrating on one aspect of the configuration at a time. For details about using these forms, see NNMi Help for each form.

The **Tenants** view contains NNMi multi-tenancy configuration information. This view is available under **Discovery** in the **Configuration** workspace. Each **Tenant** form describes one NNMi tenant and shows the nodes currently assigned to that tenant. The node assignment information is read-only.

To change the tenant or security group assignment for a node, use the **Node** form or the nnmsecurity.ovpl command.

The NNMi console views described below are available under **Security** in the **Configuration** workspace. These views contain NNMi security configuration information.

User Accounts

- Each User Accounts form describes one NNMi user and shows the user groups to which that user belongs. The membership information is read-only.
- If you are storing user group membership in a directory service, user accounts are not visible in the NNMi console.

User Groups

Each User Groups form describes one NNMi user group and shows the user accounts and security groups mapped to that user group. The mapping information is read-only.

User Account Mapping

- Each User Account Mapping form shows one user account-to-user group association.
- Changes to user account mappings do not affect the current NNMi console users. These users receive any changes the next time they sign in to the NNMi console.
- If you are storing user group membership in a directory service, user account mappings are not visible in the NNMi console.

Security Groups

Each **Security Groups** form describes one NNMi security group and shows the nodes currently assigned to that security group. The node assignment information is read-only.

Security Group Mapping

- Each Security Group Mapping form shows one user group-to-security group association.
- After initial configuration, the object access privilege associated with a security group mapping is read-only. To change the object access privilege for a security group mapping, delete that mapping and re-create it.

Command line

The nnmsecurity.ovpl command-line interface is useful for automation and bulk operations. The tool also provides reports of potential problems with the security configuration.

Many of the nnmsecurity.ovpl options support loading input data from comma-separated values (CSV) files. You can maintain configuration data in a file or system that can generate CSV output for consumption by the nnmsecurity.ovpl command. The command can also accept UUIDs generated outside of NNMi.

🛛 Тір

Because security group and tenant names do not need to be unique, specify the security group or tenant UUID as input to the nnmsecurity.ovpl command.

The following example script uses the nnmsecurity.ovpl command to create the security configuration for two user accounts and five nodes:

```
#!/bin/sh
# create two users
nnmsecurity.ovpl -createUserAccount user1 -password -role level1
nnmsecurity.ovpl -createUserAccount user2 -password -role level2
# create two user groups
nnmsecurity.ovpl -createUserGroup local1
nnmsecurity.ovpl -createUserGroup local2
# assign the user accounts to the new user groups
nnmsecurity.ovpl -assignUserToGroup -user user1 -userGroup local1
nnmsecurity.ovpl -assignUserToGroup -user user2 -userGroup local2
# create two security groups
nnmsecurity.ovpl -createSecurityGroup secgroup1
nnmsecurity.ovpl -createSecurityGroup secgroup2
# assign the new user groups to the new security groups
nnmsecurity.ovpl -assignUserGroupToSecurityGroup -userGroup local1 -securityGroup
secgroup1 -role level1
nnmsecurity.ovpl -assignUserGroupToSecurityGroup -userGroup local2 -securityGroup
secgroup2 -role level2
# assign nodes to security groups
nnmsecurity.ovpl -assignNodeToSecurityGroup -node mplspe01 -securityGroup secgroup
1
nnmsecurity.ovpl -assignNodeToSecurityGroup -node vwan router-1 -securityGroup sec
group1
nnmsecurity.ovpl -assignNodeToSecurityGroup -node vwan router-2 -securityGroup sec
group1
nnmsecurity.ovpl -assignNodeToSecurityGroup -node data center 1 -securityGroup sec
group2
nnmsecurity.ovpl -assignNodeToSecurityGroup -node mplspe03 -securityGroup secgroup
2
```

14.4.2 Configuring multi-tenancy

NNMi provides the following ways to configure multi-tenancy:

• The Tenant form in the NNMi console

This is useful for working with individual tenants.

• nnmsecurity.ovpl command-line interface

This is useful for automation and bulk operations. The tool also provides reports of potential problems with the tenant configuration.

The process of defining and configuring NNMi multi-tenancy to assign each NNMi topology object to a tenant (organization) is a cyclical process.

Note the following about configuring NNMi multi-tenancy:

• The security group that NNMi assigns to a discovered node is set by the value of the **Initial Discovery Security Group** for the tenant associated with that node.

```
14. NNMi Security and Multi-Tenancy
```

- When you use the NNMi security model without also configuring NNMi tenants, all nodes are assigned to the Default Tenant.
- When you seed a node for NNMi discovery, you can specify the tenant to which that node belongs. When NNMi discovers a node through an auto-discovery rule, NNMi assigns that node to the Default Tenant. After discovery, you can change the tenant assignment for the node.

One high-level approach to planning and configuring NNMi multi-tenancy is as follows:

- Analyze your customer requirements to determine how many tenants are required in the NNMi environment. We recommend that tenants be used only when managing multiple separate networks with a single NNMi management server.
- 2. Analyze the managed network topology to determine which nodes belong to each tenant.
- 3. Analyze the topology of each tenant to determine the groups of nodes to which NNMi users need access.
- 4. Remove the default associations between the predefined NNMi user groups and the **Default Security Group** and **Unresolved Incidents** security group.

This step ensures that users do not inadvertently obtain access to nodes they are not supposed to be managing. At this point, only NNMi administrators can access objects in the NNMi topology.

- 5. Configure the identified tenants.
 - **a.** Create the identified security groups.
 - **b.** Create the identified tenants.

For each tenant, set the **Initial Discovery Security Group** to either the **Default Security Group** or a tenant-specific security group with restricted access. This approach ensures that new nodes for the tenant are not generally visible until the NNMi administrator configures access.

6. Prepare for discovery by assigning tenants to seeds.

After discovering a group of nodes, you can change the value of the **Initial Discovery Security Group**. Using this approach limits the manual re-assignment of nodes to security groups.

- 7. After discovery completes, do the following:
 - Verify the tenant for each node and make changes as necessary.
 - Verify the security group for each node and make changes as necessary.
- 8. Configure custom user groups.

For details about configuring custom user groups, see 14.4.4 Verifying the configuration.

14.4.3 Configuring security groups

If you plan to integrate NNMi with a directory service for consolidating the storage of user names, passwords, and, optionally, NNMi user group assignments, complete that configuration before configuring NNMi security.

NNMi provides the following ways to configure security:

• The Security Wizard in the NNMi console

This wizard is useful for visualizing the security configuration. The **View Summary of Changes** tab presents a list of unsaved changes from the current wizard session. It also identifies potential problems with the security configuration.

• Forms in the NNMi console for individual security objects

These forms are useful for concentrating on one aspect of the security configuration at a time.

• nnmsecurity.ovpl command-line interface

This tool is useful for automation and bulk operations. The tool also provides reports of potential problems with the security configuration.

The process of defining and configuring NNMi security to limit users' access to objects in the NNMi topology is a cyclical process.

Note

This example moves from security groups to user accounts. For examples of configuring NNMi security from user accounts to security groups, search for *Configure Security Example* in NNMi Help.

Note the following about configuring NNMi security:

- The security group that NNMi assigns to a discovered node is set by the value of the **Initial Discovery Security Group** for the tenant associated with that node.
- When you use the NNMi security model without also configuring NNMi tenants, all nodes are assigned to the Default Tenant.

One high-level approach to planning and configuring NNMi security is as follows:

- 1. Analyze the managed network topology to determine the groups of nodes to which NNMi users need access.
- 2. Remove the default associations between the predefined NNMi user groups and the **Default Security Group** and **Unresolved Incidents** security group.

This step ensures that users do not inadvertently obtain access to nodes they are not supposed to be managing. At this point, only NNMi administrators can access objects in the NNMi topology.

- 3. Configure a security group for each subset of nodes. Remember that a given node can belong to only one security group.
 - **a.** Create the security groups.
 - **b.** Assign the appropriate nodes to each security group.
- 4. Configure custom user groups.
 - a. For each security group, configure a user group for each level of NNMi user access.
 - If you are storing user group membership in the NNMi database, no users are mapped to these user groups yet.
 - If you are storing user group membership in a directory service, set the **Directory Service Name** field for each user group to the distinguished name of that group in the directory service.

b. Map each custom user group to the correct security group. Set the appropriate object access privilege for each mapping.

5. Configure user accounts.

If you are storing user group membership in the NNMi database, do the following:

- Create a user account object for each user who is permitted to access the NNMi console. The process of configuring user accounts depends on whether you are using a directory service for NNMi console sign-in.
- Map each user account to one of the predefined NNMi user groups (for access to the NNMi console).
- Map each user account to one or more custom NNMi user groups (for access to topology objects).

If you are storing user group membership in a directory service, verify that each user belongs to one of the predefined NNMi user groups and one or more custom user groups.

6. Verify the configuration as described in 14.4.4 Verifying the configuration.

^{14.} NNMi Security and Multi-Tenancy

- 7. Maintain the security configuration.
 - Watch for nodes added to the **Default Security Group**, and move these nodes to the correct security groups.
 - Add new NNMi console users to the correct user groups.

14.4.4 Verifying the configuration

To verify that the security configuration is correct, verify each aspect of the configuration separately. This subsection describes some approaches to verifying the configuration. Other approaches are possible.

Note

NNMi provides reports of possible security configuration errors. Access these reports by choosing **Security Reports** from the **Tools** menu in the NNMi console. Alternatively, you can access the reports by using the nnmsecurity.ovpl command with the -displayConfigReport option specified.

Verifying security group-to-node assignments

Verify that each node is assigned to the correct security group by using one of the following methods:

- Sort the Nodes or Nodes (All Attributes) inventory view by security group, and then examine the groupings.
- Use the nnmsecurity.ovpl command with the -listNodesInSecurityGroup option specified.

Verifying user group-to-security group assignments

Verify which user groups are mapped to each security group by using one of the following methods:

- Sort the **Security Group Mapping** view by user group or security group, and then examine the groupings. Also verify the object access privilege for each mapping.
- On the Map User Groups and Security Groups tab of the Security Wizard, select one user group or security group at a time to see the current mappings for that object.
- Use the nnmsecurity.ovpl command with the -listUserGroupsForSecurityGroup option specified.

Verifying that each user has NNMi console access

For NNMi console access, ensure that each user is assigned to one of the predefined NNMi user groups:

- NNMi Administrators
- NNMi Level 1 Operators
- NNMi Level 2 Operators
- NNMi Guest Users

All other user group assignments provide access to objects in the NNMi database.

Users without NNMi console access are listed on the **View Summary of Changes** tab of the **Security Wizard**. The **Security Reports** menu item under the **Tools** menu and the -displayConfigReport usersWithoutRoles option to the nnmsecurity.ovpl command also provide this information.

Verifying user-to-user group assignments

Verify user group membership by using one of the following methods:

- Sort the User Account Mapping view by user account or user group, and then examine the groupings.
- On the Map User Accounts and User Groups tab of the Security Wizard, select one user account or user group at a time to see the current mappings for that object.

• Use the nnmsecurity.ovpl command with the -listUserGroups and -listUserGroupMembers options specified.

Verifying tenant-to-node assignments

One approach to verifying that each node is assigned to the correct tenant is to sort the **Nodes** or **Nodes** (All Attributes) inventory view by tenant, and then examine the groupings.

Verifying current user settings

To verify the NNMi console access for the currently logged-on user, click **Help**, and then click **System Information**. The **User Information** section on the **Product** tab lists the following information for the current NNMi session:

- User name as defined for the user account in the NNMi database or the accessed directory service.
- NNMi role, which corresponds to the most privileged of the predefined NNMi user groups to which the user is mapped (NNMi Administrators, NNMi Level 2 Operators, NNMi Level 1 Operators, or NNMi Guest Users). This mapping determines which actions are available within the NNMi console.
- User groups mapped to this user name. This list includes the predefined NNMi user group that sets the NNMi role and any other user groups that provide access to objects in the NNMi database.

14.4.5 Exporting the NNMi security and multi-tenancy configuration

The table below describes the configuration areas (available with the nnmconfigexport.ovpl -c command) for exporting the NNMi security and multi-tenancy configuration. These export areas are beneficial for maintaining the configuration across multiple NNMi management servers, especially in a Global Network Management environment.

Configuration area	Description
account	Exports user accounts, user groups, and user account-to-user group mappings. Useful for sharing user definitions across multiple NNMi databases.
security	Exports tenants and security groups. Useful for sharing security definitions across multiple NNMi databases. Importing this information creates new objects and updates existing objects but does not delete objects not included in the current export. Therefore, this option is safe to use with an NNMi database containing locally-defined objects.
securitymappings	Exports user group-to-security group mappings. For a complete export of the security and multi-tenancy configuration, perform a concurrent export of the account, security, and securitymappings configuration areas.

Table 14-5: NNMi security and multi-tenancy configuration export areas

14.5 Defining NNMi security and multi-tenancy in global network management

In a global network management (GNM) environment, a node's tenant is set on the NNMi management server that manages that node. The tenant UUID for a given node is the same on each global and regional manager in the GNM environment.

A node's security group is set on each NNMi management server whose topology contains that node. Thus, user access to objects in the topology is configured separately on each NNMi management server in the GNM environment. The global and regional managers might use the same or different security group definitions.

If you want user access to be similar on the global manager and regional managers, you can employ some configuration tricks, but you probably cannot completely avoid custom configuration on each NNMi management server.



• Define all tenants and security groups on the global manager. Use nnmconfigexport.ovpl-c security to export the tenant and security group definitions. On each regional manager, use nnmconfigimport.ovpl to import the tenant and security group definitions. Alternatively, you can use the nnmsecurity.ovpl command to create tenants and security groups with the same UUIDs as on another NNMi management server. Following this recommendation ensures that each tenant and security group has the same UUID within the GNM environment.

This best practice becomes a required part of the configuration if users will be launching NPS reports from the global manager.

Tenant UUIDs must be unique, but tenant names can be reused. NNMi considers two tenants with the same name and different UUIDs to be two distinct tenants with no shared configuration.

• If you are setting up one regional manager per organization, all nodes on a regional manager can be in a single tenant. However, configure a unique tenant on each regional manager to ensure separation of the topology data on the global manager.

Incidents forwarded from a regional manager to a global manager might include some additional custom incident attributes (CIAs) to convey security and tenant information.

If the incident's source object belongs to a tenant other than the Default Tenant, the forwarded incident contains the following CIAs:

cia.tenant.name

cia.tenant.uuid

If the incident's source object belongs to a security group other than the **Default Security Group**, the forwarded incident contains the following CIAs:

cia.securityGroup.name

cia.securityGroup.uuid

14.5.1 Initial configuration of security and multi-tenancy in global network management

After a global network management (GNM) is first configured, the regional manager updates the global manager with information about the nodes in the regional topology (according to the GNM configuration).

^{14.} NNMi Security and Multi-Tenancy

Topology synchronization with the Default Tenant only

For GNM environments with custom security groups and the Default Tenant, on the global manager all nodes managed remotely are added to the global manager topology with the following configuration:

- Default Tenant
- The security group that is set as the Initial Discovery Security Group for the Default Tenant

Topology synchronization with custom tenants

For GNM environments with custom security groups and custom tenants, on the global manager all nodes managed remotely are added to the global manager topology with the UUID of the tenant assigned to the node. If that tenant UUID does not exist on the global manager, the GNM processes create that tenant in the NNMi configuration of the global manager as follows:

- The tenant UUID is the same value as on the regional manager.
- The tenant name is the same value as on the regional manager.
- The value of the **Initial Discovery Security Group** is set to the security group with the same name as the tenant. (NNMi creates this security group if it does not already exist on the global manager.)

As the node is added to the topology on the global manager, it is assigned to the **Initial Discovery Security Group** for the tenant UUID as configured on the global manager. That is, the security group association on the global manager is independent of the security group association on the regional manager.

О Тір

The following are suggestions for simplifying security configuration on the global manager:

- Maintain a spreadsheet or other record of the nodes managed by each regional manager. For each node, note the expected security group on the regional manager and that on the global manager. After GNM configuration completes, use the nnmsecurity.ovpl command to verify and update the security group assignments.
- If the GNM environment will include multiple regional managers updating a single global manager, enable the GNM configuration from one regional manager at a time to the global manager.
- If appropriate, you can change the value of the **Initial Discovery Security Group** of the Default Tenant (or a custom tenant) before adding each regional manager to the GNM configuration. Note that this approach can have mixed results if new nodes are being added to the topology on the previously configured regional managers.
- Before enabling GNM, on the global manager set the **Initial Discovery Security Group** of each tenant used on the regional manager to be a private security group that operators cannot access. An administrator on the global manager then needs to explicitly move the nodes to the appropriate security groups for other NNMi console operators.

14.5.2 Effects of security and multi-tenancy assignment on a global network management

The following table describes how changes to a node's tenant or security group assignment on a regional manager affect the global manager.

^{14.} NNMi Security and Multi-Tenancy

Table 14-6: Global manager impact of configuration changes on a regional manage

Action	Effect
On the regional manager, assign a node to a different tenant.	The node on the global manager is changed to be assigned to the different tenant. If this tenant UUID does not exist on the global manager, it is created.
On the regional manager, assign a node to a different security group.	No change on the global manager. The NNMi administrator can choose to replicate the change manually.
On the regional manager, change the configuration (name, description, or initial discovery security group) of a tenant.	No change on the global manager. The NNMi administrator can choose to replicate the change manually.
On the regional manager, change the configuration (name or description) of a security group.	No change on the global manager. The NNMi administrator can choose to replicate the change manually.

^{14.} NNMi Security and Multi-Tenancy

15

Global Network Management

This chapter describes how to manage a global network.

To use the global network management feature, you must ensure that the versions and revisions of all NNMi management servers composing the global network management are the same. Their update versions can be different.

^{15.} Global Network Management



Suppose you have NNMi deployed on multiple NNMi management servers in several geographic locations. You have these NNMi management servers discovering and monitoring the network to meet your discovery and monitoring needs. Using these existing NNMi management servers and configurations, you can designate specific NNMi management servers as global managers to display combined node object data without additional discovery or monitoring configuration changes.

The NNMi global network management feature enables multiple NNMi management servers to work together while managing different geographic areas of the network. You designate specific NNMi management servers as global managers to display combined node object data from two or more regional managers.

The NNMi global network management feature offers the following benefits:

- Provides a central big-picture view of your corporate-wide network from the global manager
- Easy to set up:
 - Each regional manager administrator specifies node object data from all nodes or from a specific group of nodes for inclusion at the global manager level.
 - Each global manager administrator specifies which regional managers are allowed to contribute information.
- Generates and manages incidents independently on each server (generated within the context of the topology available on each server)

For details, see NNMi's Global Network Management Feature (NNMi Advanced) in NNMi Help.

Each group of dynamic network address translation (NAT) or dynamic port address translation (PAT) or dynamic network address and port translation (NAPT) requires an NNMi regional manager, in addition to a tenant that is unique within the entire NNMi global network management configuration. For details, see 13. Managing Overlapping IP Addresses in a NAT Environment and NNMi Help.

^{15.} Global Network Management
15.3.1 Monitoring networks at multiple sites continuously

If your information technology group manages network equipment located at multiple sites on a 24/7 basis, your group can use NNMi's global network management feature to observe combined topology and incident views.

15.3.2 Monitoring selected critical devices

To view device status and incidents for critical devices located at multiple locations from one NNMi management server, you must configure forwarding filters on the regional managers. Doing so enables you to select the node object data you want the regional managers to send to the global manager. For example, you can set up forwarding filters on the regional managers so that only information about critical devices is forwarded to the global manager.

15.3.3 Considering licensing

You must purchase and install NNMi Advanced licenses on the NNMi management server you plan to use as the global manager. NNMi management servers do not need NNMi Advanced licenses to function as regional managers.

You can use the global network management feature to limit the new licenses you need on the global manager. For example, if your information technology group needs to monitor critical equipment located at multiple sites, you can configure a forwarding filter on the regional managers to make sure you only forward information about critical devices to the global manager. This enables you to make wise use of your NNMi investment and maintain control over your use of the license capacity you have on the global manager.

If you have increased the NNMi licenses for the regional managers such that the total number of licensed nodes is larger than the NNMi Advanced licenses on the global manager, the global manager will not have been able to maintain a complete inventory of all nodes in all regions. To synchronize the global manager with all of the regional managers so that the global manager can find and create the nodes it skipped due to insufficient licenses, you must purchase and install enough NNMi Advanced licenses on the global manager to meet or exceed the total number of licenses you have installed on the regional managers.

After you have enough licenses installed, do one of the following:

• Wait for all the configured rediscovery intervals on all the regional managers to elapse so that all nodes in all regions become rediscovered. Once the regional managers have rediscovered all nodes in all regions, they send this rediscovered node information to the global manager.

The global manager receives this node information and creates global nodes for all nodes in each region.

• Run the nnmnoderediscover.ovpl -all script on each regional manager.

Note

The second option results in both a lot of traffic on your network and consumption of a lot of NNMi resources from the entire set of NNMi managers. This option is not as resource intensive as the initial NNMi discovery, but it is similar to doing an initial discovery. The best approach is to space the running of the script for each region by some amount of time or by waiting for the current regional manager's workload to drop to normal before starting the next regional manager's rediscovery.

^{15.} Global Network Management

15.4 Practical global network management examples

The example in the figure below shows a company that has two operating sites in different geographic locations. The company headquarters is located in a third geographic area. NNMi management servers are functioning at all three locations.

From a network perspective, the IT group at corporate headquarters needs to monitor the local network equipment at the headquarters as well as critical network equipment located at regional sites 1 and 2. The IT groups at regional sites 1 and 2 need to monitor the local critical network equipment located at their respective sites.





15.4.1 Reviewing the requirements

Suppose the NNMi management servers at corporate headquarters, regional site 1, and regional site 2 manage several routers and switches located at their individual sites. For this example, refer to the NNMi management servers as global1, regional1, and regional2, respectively. Suppose you configured these NNMi management servers to discover and monitor critical switches and routers located at their own locations. In such a case, there is no need to reconfigure discovery for the NNMi management servers at any of these sites to use the global network management feature.

Note

During global network management configuration, you might be tempted to use the nnmbackup.ovpl script to back up one NNMi management server, then use the nnmrestore.ovpl script to restore this backup to a second NNMi management server, then connect both of these NNMi management servers to a regional NNMi management server. Do not do this. Placing the backup data from one NNMi management server onto a second NNMi management server means that both servers have the same database UUID. After you restore NNMi on the second NNMi management server, you would need to uninstall NNMi from the original NNMi management server.

The IT group at the corporate site wants to monitor critical equipment located at regional sites 1 and 2, but they do not want to manage all devices. The following table summarizes the monitoring needs.

^{15.} Global Network Management

Table 15-1:	Network requirements	for global	network management
		5	J

Site	NNMi management server	Critical switches	Regional equipment to manage
Corporate headquarters	global1	15 Model 3500yl HP ProCurve switches	All Model 3500yl HP ProCurve switches at all regional sites
Regional site 1	regional1	15 Model 3500yl HP ProCurve switches	Not applicable
Regional site 2	regional2	15 Model 3500yl HP ProCurve switches	Not applicable

To summarize, the NNMi management server, global1, monitors the corporate headquarters. Two NNMi management servers, regional1 and regional2, monitor each of the regional sites. It is essential to be able to view from corporate headquarters the incidents and device information for the Model 3500yl ProCurve switches located at regional sites 1 and 2. Suppose that, for this example, regional1 and regional2 both manage several common switches located at regional site 1.

(1) Regional manager and global manager connections

When you configure global network management connections, consider the following information:

- NNMi enables you to configure more than one global manager to communicate with a regional manager. For example, if you need a second global manager, global2, to communicate with regional1, NNMi enables you to configure both global1 and global2 to communicate with regional1. For details, see the *Release Notes*.
- Global network management works with one connection layer. For example, the examples in this chapter discuss one connection layer, globall communicating with regional1 and globall communicating with regional2. Do not configure NNMi for multiple connection levels. For example, do not configure globall to communicate with regional1, then configure regional1 to communicate with regional2. The global network management feature is not designed for this type of three-layer configuration.
- Do not configure two NNMi management servers to communicate both ways with each other. For example, do not configure globall to communicate with regional1, then configure regional1 to communicate with global1.

15.4.2 Initial preparation

(1) Port availability: Configuring the firewall

For the global network management feature to function properly, verify that certain well-known ports are open for TCP access from globall to regionall and regional2. The NNMi installation script sets http port 80 and https port 443 by default, but you can change this value during installation.

Note

In the example discussed in this subsection, globall establishes TCP access to regional1 and regional2. Firewalls are usually configured based on the server initiating the connection. After global1 establishes the connection to regional1 and regional2, traffic flows in both directions.

Edit the following file to see the current values or to make changes to the port configuration:

^{15.} Global Network Management

- Windows: %NNM CONF%\nnm\props\nms-local.properties
- Linux: \$NNM_CONF/nnm/props/nms-local.properties

The following table shows the well-known ports that need to be accessible.

Table 15-2:	Required	accessible	sockets
			00011010

Security	Parameter	TCP port
Non-SSL	nmsas.server.port.web.http	80
	nmsas.server.port.hq	4457
SSL	nmsas.server.port.web.https	443
	nmsas.server.port.hq.ssl	4459

(2) Configuring Certificates

If you plan to use the global network management feature with a secure communication protocol between global1 and the two regional NNMi management servers (regional1 and regional2), you must configure the certificates. During NNMi installation, the NNMi installation script creates a self-signed certificate on the NNMi management server so it can identify itself to other entities. Configure the NNMi management servers you plan to use with the global network management feature with the correct certificates. Complete the steps shown in 10.3.7 Working with Certificates in Global Network Management Environments.

NNMi management servers upgraded to the version 11-50

If you are working with a set of NNMi management servers where some management servers were upgraded to NNMi 11-50 from an older version of NNMi and some management servers have newly installed instances of NNMi 11-50 you must perform some additional configuration tasks before configuring GNM.

Prior to the version 11-50, NNMi used to provide a Java KeyStore (JKS) repository to store certificates. NNMi 11-50 introduces a Public Key Cryptography Standards (PKCS) #12 repository to store certificates. The new PKCS #12 filebased certificate management technique is available for use as soon as you install a new instance of NNMi 11-50 on a system.

However, when you upgrade an older version of NNMi to the version 11-50, the PKCS #12 file-based certificate management does not immediately come into effect and NNMi continues to use the JKS repository for certificate management.

Before configuring GNM in this kind of environment, make sure that all upgraded NNMi management servers are configured to use the PKCS #12 file-based certificate management technique by following the instructions in 10.2 Configuring an Upgraded NNMi Environment to Use the New Keystore.

(3) NNMi management server sizing considerations

This example assumes you plan to use existing NNMi management servers in a global network management configuration.

For specific information about the size of server you need for NNMi, see the Release Notes.

(4) Synchronizing system clocks

It is important that you synchronize the NNMi management server clocks for global1, regional1, and regional2 before you connect these servers in a global network management configuration. All NNMi management

^{15.} Global Network Management

servers in your network environment that participate in global network management (global managers and regional managers) or single sign-on (SSO) must have their internal time clocks synchronized in universal time. Use a time synchronization program, such as the Linux Network Time Protocol Daemon (NTPD) tool or one of the available Windows operating system tools. For details, see *Clock Synchronization Issues (SSO/Global Network Management)* or *Troubleshoot Global Network Management* in NNMi Help and 15.11.2 Clock synchronization.

Note

NNMi opens a warning message at the bottom of the NNMi console if there is a connection problem with a regional manager, such as a server clock synchronization issue.

(5) Using the application failover feature with self-signed certificates in global network management

If you plan to use the global network management feature using self-signed certificates in an application failover configuration, you must complete some additional steps.

(6) Using self-signed certificates in global network management

If you plan to use the global network management feature using self-signed certificates, you must complete some additional steps. For details, see 10.3.7 Working with Certificates in Global Network Management Environments.

(7) Using a Certificate Authority in global network management

If you plan to use the global network management feature using a Certificate Authority, you must complete some additional steps. For details, see 10.3.7 Working with Certificates in Global Network Management Environments.

(8) Listing the critical equipment you want to monitor

Compile a list of the equipment managed by each regional manager and monitored from the global manager. For example, compile a list of the equipment managed by regional1 and regional2 that you want to monitor from global1. You use this information in a forwarding filter. For details, see 15.5 Configuring forwarding filters on the regional managers.

Carefully consider the possible outcomes of limiting the information forwarded to globall from regional1 and regional2. Below are some points to consider during your planning:

- Be careful not to exclude too many devices, as globall needs a complete topology from regional1 and regional2 to do a complete analysis to generate accurate incidents.
- Excluding non-critical devices helps reduce system performance costs on global1.
- Excluding non-critical devices helps improve the solution's overall scalability and reduce the network traffic required by NNMi.

(9) Reviewing the global and regional managers' management domains

Review the global and regional managers' management domains to help determine the information you want to forward from the regional managers to the global manager.

In our example, NNMi management servers global1, regional1, and regional2 manage their own sets of nodes. Later in this example, you configure regional1 and regional2 to forward to global1 information about equipment they manage.

Use the procedure below to understand the equipment that global1, regional1, and regional2 currently monitor. This will assist you in selecting the critical equipment you want regional1 and regional2 to forward to global1.

For this example, complete the following steps to review this information:

- 1. Point your browser to global1's NNMi console.
- 2. Sign in.
- 3. Click Inventory workspace.
- 4. From here you can review the discovered inventory globall currently monitors.
- 5. Point your browser to regional1's NNMi console.
- 6. Sign in.
- 7. Click Inventory workspace.
- 8. Review the nodes that regional1 monitors and compile a list of the devices you want to monitor from global1.
- 9. Point your browser to regional2's NNMi console.
- 10. Sign in.
- 11. Click **Inventory** workspace.

12. Review the nodes that regional2 monitors and compile a list of the devices you want to monitor from global1.

(10) Reviewing NNMi Help topics

To review all the Help topics related to global network management, complete the following steps:

- 1. From NNMi Help, click Search.
- 2. In the Search field, type Global Network Management.
- 3. Click Search.

This search will result in more than 50 topics related to global network management.

^{15.} Global Network Management

15.5 Configuring forwarding filters on the regional managers

In this example, globall communicates with both regional1 and regional2. To control the node object data you want global manager global1 to receive from regional managers regional1 and regional2, configure forwarding filters on regional1 and regional2.

15.5.1 Configuring a forwarding filter to limit forwarded nodes

The example creates a node group to enable regional1 to forward to global1 only node information for Model 3500yl ProCurve Switches. To create a new node group and set this limitation, complete the following steps:

1. From Configuration > Object Groups for regional1 on the NNMi console, click Node Groups.

15. Global Network Management

	Network	k Node I	Manager i
--	---------	----------	-----------

Lill Dashboards
هد Incident Management
📥 Topology Maps
Monitoring
A Troubleshooting
E Inventory
Management Mode
R Incident Browsing
S Integration Module Configuration
✤ Configuration
Communication Configuration
Discovery
Monitoring
Incidents
Status Configuration
i Global Network Management
User Interface
🕨 🛅 Security
🕨 🖿 MIBs
Device Profiles
🕶 🗁 Object Groups
Node Groups
Interface Groups
E RAMS Servers

<u>F</u> ile <u>V</u>	íew	Tools	Actions	He	elp	Us
Node Grou	ups 🗙					
2 *		C 💈	• •	Í	•	
Status	▲ Name	Ð			Add to View Filter List	Calcula Status
Ø	Importa	nt Nodes			~	-
0	Microso	ft Windows	Systems		*	-
0	Neighbo	or Connectio	ons Filter		-	-
0	Network	king Infrastru	ucture Device	s	~	-
\oslash	Non-SN	IMP Devices	s		~	-
0	Routers				~	-
Ø	Switche	s			~	-
0	VMware	e ESX Hosts	5		~	-
0	Virtual I	Machines			~	-

2. Click New.

15. Global Network Management

Network Node Manager i	<u>F</u> ile	⊻iew	Tools	Actions	<u>H</u> elp	
الله Dashboards معنى Incident Management	Node Gro	oups 🗙				
	C (*) 🖬	C 💈	• •		
Monitoring	Status	▲ Name	e		Add to View	Calc State
A Troubleshooting	0	Incode	at Madaa		Filter List	
	0	importa	ntivodes		•	-
	0	Microso	ft Windows	s Systems	~	-
र्श्र Incident Browsing	0	Neighbo	or Connecti	ions Filter	-	-
S Integration Module Configuration	0	Network	king Infrast	ructure Device	s 🗸	-
🗲 Configuration	0	Non-SN	IMP Device	es	*	-
Communication Configuration	0	Routers)		~	-
Discovery	\oslash	Switche	S		~	-
Monitoring	0	VMware	e ESX Host	ts	~	-
Incidents	0	Virtual M	Machines		~	-
Status Configuration						
Global Network Management						
User Interface						
Security						
MIBs						
Device Profiles						
🗢 🗁 Object Groups						
Node Groups						
Interface Groups						
E RAMS Servers						

Note

Although this example explains how to create a new node filter, and then use it to create forwarding filters from regional1 and regional2, you can use any existing filter as a model to be modified

to serve as a forwarding filter from a regional NNMi management servers to the global NNMi management server.

You can create a container node group that contains no devices or filters of its own, then use this node group to specify child node groups. Using this approach, you can forward node object data to global NNMi management servers using one container node group.

3. Type globall in the Name field as the filter name and enter in the Notes field any notes you need about the filter you are creating.



4. Click the New icon on the Device Filters tab to open a Node Device Filter form.

^{15.} Global Network Management

Node Groups at Node Group * 30						
6 6 5 5 5						
Basics Name global1	∧ Device F	ilters	Additional Filters	Additional Nodes	Child Node Groups	Status
Calculate Status Status No Status Add to View Filter List	Device Fi Family, or of the Dev	Iters enab r Device P vice Filters	le you to determin rofile. If you config and pass any Ac	e Node Group member gure multiple Device Fil diftional Filters to belon	s by Device Category, ters, Nodes must match g to this Node Group.	vendor, at least one
Pass Procurve switches to global1	Devi	ice Categ	Device Vendor	Device Family	Device Model	
You can filter Node Groups using Device Filters, Additional Filters, Additional Nodes, and Child Node Groups. If you use Device Filters and Additional Filters, Nodes must match at least one Device Filter and the Additional Filters specifications to belong to this Node Group. Nodes that are specified as Additional Nodes and Child Node Groups always are members of this Node Group. See Help \rightarrow Using the Node Group form. To test your Node Group definition, select File \rightarrow Save, then Actions \rightarrow Node Group Details \rightarrow Preview Members (Current Group Only).						
 NNM iSPI Performance 						
Used by NNM iSPI Performance for Metrics and NNM iSPI for Traffic.						
	~			Total: 0 Selected: 0	Filter: OFF Auto refree	sh: OFF
		_				
Analysis - Summary - No Objects Selected						

- 5. Using the pull-down, select Switch-Router for Device Category, Hewlett-Packard for Device Vendor, and HP ProCurve 3500 Fixed-port Switch for Device Family.
- 6. Using the pull-down, click **Quick Find** to open a **Device Profile** form.

^{15.} Global Network Management

Switch-Router -
Hewlett-Packard -
HP ProCurve 2620 Fixed-port Switch 👻
-
Show Analysis

7. Find and select the profile for the HP ProCurve 3500yl switch, then click **OK**.

^{15.} Global Network Management

Quick Find

▲ Device Model	SNMP Object ID	OUI	Device Family	Device Vendor	De
hpLanProbeFDDI	.1.3.6.1.4.1.11.2.3.7.6.6.1		HP LAN Probe	I Hewlett-Packar	~
hpLanProbellI	.1.3.6.1.4.1.11.2.3.7.6.5.1		HP LAN Probe	Hewlett-Packar	
hpLaserJetP2015	.1.3.6.1.4.1.11.1		HP Printers	Hewlett-Packar	
hpMidLevelMgr	.1.3.6.1.4.1.392.1.1		HP LAN Analyzer Age	e NetMetrix	
hpNAC800	.1.3.6.1.4.1.11.2.3.7.11.73		HP Service Appli	a 🞲 Hewlett-Packar	
hpONEServices	.1.3.6.1.4.1.11.2.3.7.11.50		HP Switch Services M	Hewlett-Packar	
hpP2910al-48G-PoE+	.1.3.6.1.4.1.11.2.3.7.11.85		HP 2910 al Switch Se	e 🞲 Hewlett-Packar	
hpPortModuleJ4821A	.1.3.6.1.4.1.11.2.3.7.11.17.		HP Switch Module	Mewlett-Packar	
hpPowerAgent	.1.3.6.1.4.1.392.1.0		HP LAN Analyzer Age	e NetMetrix	
hpProCurve10T100THu	.1.3.6.1.4.1.11.2.3.7.5.22		HP ProCurve Hubs	Mewlett-Packar	
hpProCurve10T100THu	.1.3.6.1.4.1.11.2.3.7.5.23		HP ProCurve Hubs	Mewlett-Packar	
hpProCurve2610-48-PV	.1.3.6.1.4.1.11.2.3.7.11.79		HP 2600 Switch S	S 🞲 Hewlett-Packar	
hpProCurve2620-24-Po	.1.3.6.1.4.1.11.2.3.7.11.13		HP ProCurve 2620 F	i: 🞲 Hewlett-Packa	
hpProCurve4202vl-68	.1.3.6.1.4.1.11.2.3.7.11.71		HP ProCurve 420	😥 Hewlett-Packar	
hpProCurve7000	.1.3.6.1.4.1.11.2.14.11.7.1		HP ProCurve 700	Hewlett-Packar	
hpProCurve8100fl_8108	.1.3.6.1.4.1.11.2.14.11.8.1.		HP ProCurve 810	😥 Hewlett-Packar	
hpProCurve8116fl	.1.3.6.1.4.1.11.2.14.11.8.1		HP ProCurve 810	Hewlett-Packar	
hpProCurveA)P10A)g	.1.3.6.1.4.1.11.2.14.11.6.4		HP ProCurve Acc	😥 Hewlett-Packar	-
hpProCurveAP420	.1.3.6.1.4.1.11.2.14.11.6.4		HP ProCurve Acc	Hewlett-Packar	
hpProCurveAP530	.1.3.6.1.4.1.11.2.14.11.6.4		HP ProCurve Acc	😥 Hewlett-Packar	
hpProCurveMSM310	.1.3.6.1.4.1.8744.1.20		HP ProCurve Acc	😥 Hewlett-Packar	
hpProCurveMSM310R	.1.3.6.1.4.1.8744.1.43		HP ProCurve Acc	Mewlett-Packar	
hpProCurveMSM313	.1.3.6.1.4.1.8744.1.16		HP ProCurve Acc	Hewlett-Packar	~
<				>	
Updated: 8/25/16 04:14:	18 PM Total: 104	06 S	Selected: 1 Filter: OF	F Auto refresh: OFI	F
			Clear	Cancel	

8. Click Save and Close for each configuration form.

×

() Changes are no	ot committed until the top-level form is saved!
-	
Device Category	Switch-Router -
Device Category Device Vendor	Switch-Router Hewlett-Packard
Device Category Device Vendor Device Family	Switch-Router Hewlett-Packard HP ProCurve 2620 Fixed-port Switch

- 9. To test this filter, select global1.
- 10. From the Actions menu, choose Node Group Details, and then click Show Members.

Network Node Manager i	<u>File</u>	/iew <u>T</u> ools Ag	tions <u>H</u> e	ыр	User N	Name: administrator	NNMi Role: Admir
ull Dashboards		13	Maps	•			
ه Incident Management	Node Gro	ups 🗙	Node Group	Details >	Previe	w Members (Current (Group Only)
📥 Topology Maps	2 *	🖬 🔉 💈	Managemer	nţ Mode ►	Show	Members (include Chi	ld Groups)
				Add to	Show Charles	All Incidents	
A Troubleshooting	Status	▲ Name		Filter List	: Status	All Open incidents Details	
Inventory	0	Important Nodes		~	- maintaine	Aug 24, 2016 12:16:	21 PN Important Noc
A Management Mode	0	Microsoft Windows Sys	tems	~	-	Aug 24, 2016 12:16:	21 Ph Any system ru
R Incident Browsing	Ø	Neighbor Connections	Filter	-	-	Aug 24, 2016 12:16:	21 PN Used as a filte
% Integration Module Configuration	0	Networking Infrastructu	re Devices	~	-	Aug 24, 2016 12:16:	21 Ph Networking In
F Configuration	0	Non-SNMP Devices		~	-	Aug 24, 2016 12:16:	21 PN Nodes which I
Communication Configuration	0	Routers		~	-	Aug 24, 2016 12:16:	21 PN Includes node
Discovery	0	Switches		~	-	Aug 24, 2016 12:16:	21 PN Includes node
Monitoring	0	VMware ESX Hosts		~	-	Aug 24, 2016 12:16:	21 PN VMware ESX
Incidents	Ø	Virtual Machines		~		Aug 24, 2016 12:16:	21 PN Virtual Machir
Status Configuration Global Network Management	0	global1		*	*	Aug 29, 2016 3:09:2	2 PM Pass Procurve

11. Notice that NNMi has already discovered one HP ProCurve 3500yl switch. This indicates that the filter you created is finding the switch models you configured it for. The next step is to use this node filter you just created to configure the forwarding filter.

Node Group	ps 🗶	Nod	es 🗙									
12	0	1	4	۳.,	Î	0	lobal1			•	K	4
Status	Dev 🔻	Name		Hostna	ame	Ma	System	Location	Device	Profi	le	5
0	👬 w	answ-1		wansw-	-1	10	5 upper	east compu	hpProCu	urve2	2620-24	

12. From regionall's Configuration workspace on the NNMi console, click Global Network Management.

Ne	etwo	ork Node Manager i
ad	Das	hboards
م د ر	Incid	dent Management
*	Тор	ology Maps
Ţ	Mon	itoring
▲	Τιοι	ıbleshooting
-	Inve	ntory
Ω	Man	agement Mode
R	Incid	dent Browsing
%	Inte	gration Module Configuration
æ	Con	figuration
		Communication Configuration
►		Discovery
►		Monitoring
		Incidents
		Status Configuration
		Global Network Management
►		User Interface
►		Security
►		MIBs
	=	Device Profiles
		Object Groups
	▦	RAMS Servers

13. Click the Forwarding Filter tab.

Global Network Management 🗶					
C D D C					
•	Forwarding Filter	Regional Manager Connections Custom Attribute Replication			
For information about this form, click here.	· Node Group F	Filter			
Caution: The system clocks must be synchronized for all NNMi management servers involved in your Global Network Management environment (Regional Managers and Global Managers). For more information, click here.	Optional. Regional Managers only: If no Node Group is specified, all Node object data is forwarded from this Regional NNMi management server to all Global NNMi management servers in your network environment. Designate a Node Group to limit the amount of Node object data this Regional Manager fowards to Global Managers. For more information, click here.				
	Node Group		- 🕼 -		
14. Click Quick Find.					

•	Forwarding Filter Regional Manager Connections Custom Attribute Replication	
For information about this form, click here.	Node Group Filter	
Caution: The system clocks must be synchronized for all NNMi management servers involved in your Global Network Management environment (Regional Managers and Global Managers). For more information, click here.	Optional. Regional Managers only: If no Node Group is specified, all Node object data is forwarded from this Regional NNMi management server to all Global NNMi management servers in your network environment. Designate a Node Group to limit the amount of Node object data this Regional Manager fowards to Global Managers. For more information, click here.	•
	Show Analysis C Quick Find Open * New	

15. Select the global1 filter, then click OK.

^{15.} Global Network Management

Quick Find

▲ Name				
Important Nodes				
Microsoft Windows Systems				
Neighbor Connections Filter				
Networking Infrastructure Devices				
Non-SNMP Devices				
Routers				
Switches				
VMware ESX Hosts				
Virtual Machines				
global1				
Updated: 8/25/16 05:20:30 PM	Total: 10	Selected: 1	Filter: OFF	Auto refresh: OFF
		Clear	ОК	Cancel

16. Click Save and Close.

×

^{15.} Global Network Management

JP1/Network Node Manager i Setup Guide

Global Network Management * 🗴				
☞ 🖻 🔞 ≎				
· ·	Forwarding Filter	Regional Manager Connections	Custom Attribute Replication	
For information about this form, click here.	Node Group I	Filter		
Caution: The system clocks must be synchronized for all NNMi management servers involved in your Global Network Management environment (Regional Managers and Global Managers). For more information, click here.	Optional. Regional N Regional NNMi man Designate a Node G Managers. For more Node Group	Anagers only: If no Node Group is a agement server to all Global NNMi n iroup to limit the amount of Node obj information, click here.	pecified, all Node object data is fr nanagement servers in your netwo ect data this Regional Manager fo	warded from this ork environment. wards to Global

This completes the task of setting up a forwarding filter on regional1. Once you complete steps 1 through 16 for regional2, you will be ready to connect global1 to regional1 and regional2 as described in 15.6 Connecting a global manager with a regional manager.

^{15.} Global Network Management

15.6 Connecting a global manager with a regional manager

In this example, regional1 and regional2 both manage several switches in common.

To forward this common switch information to global1 from regional1, you need to set up the required connection.



To accomplish this, you must connect global1 to regional1 before connecting it to regional2. By using this connection sequence, global1 considers regional1 to be the NNMi management server monitoring these common switches. global1 also ignores information about these common switches that it receives from regional2.

🖹 Note

We recommend that you use this feature on a small scale to better understand how it works, then expand it to meet your network management needs.

To connect global1 first to regional1, then to regional2, complete the following steps:

1. First, synchronize the NNMi management server clocks for global1, regional1, and regional2 before you connect these servers in a global network management configuration.

For details, see Clock Synchronization Issues (SSO/Global Network Management) in NNMi Help.

Note

NNMi displays a warning message if there is a connection problem with a regional manager, such as a server clock synchronization issue.

2. Set up a connection from global1 to regional1.

a. From the global1 NNMi console, click Global Network Management in the Configuration workspace.

Network Node Manager i

<u>lahl</u>	Das	hboards
م د ه	Incid	lent Management
*	Тор	ology Maps
Ģ	Mon	itoring
A	Тгоц	ıbleshooting
101	Inve	ntory
Ω	Man	agement Mode
ର	Incid	lent Browsing
æ	Integ	gration Module Configuration
æ	Con	figuration
		Communication Configuration
►		Discovery
►		Monitoring
►		Incidents
		Status Configuration
		Global Network Management
•		User Interface
►		Security
►		MIBs
	▦	Device Profiles
►		Object Groups
	▦	RAMS Servers

b. Click Regional Manager Connections.



c. Click the 💌 New icon to create a new regional manager.

Global Network Management 🗶				
•	Forwarding Filter	Regional Manager Connections	Custom Attribute Replication	
For information about this form, click here.	-			
Caution: The system clocks must be synchronized for all NNMi management servers involved in your Global Network Management environment (Regional Managers and Global Managers). For more information, click here.	Optional. Global Mai For more information Tip: Consider loggin Node object data.	nagers only: List the Regional Manaş n, click here. g into each Regional Manager and c	pers from which you want to rece onfiguring a Forwarding Filter to I	ive Node object data.
	ଟ 🔹 🖷	C ∽ Ђ 🔒	₩ ← 0-00	0 🗲 🕅 🔁
	Name	Connection State	Desc	ription

- d. Add name and description information for regional1.
- e. Click the Connection tab.

f. Click the ***** New icon.

Globel Network Management 3: Regional Manager * #		
C 🖸 🗓 🕽 🗯		
• Basics	Connection e	
* Name (regional1	•	
Connection State	Configure either HTTP or HTTPS communication with this Regional N	INMi management server.
Description d	🛞 🗃 0 5 🕯	K ← 0.000 → N
Corporate Headquarters	Orr Hostname Use HTTP Enc (S) Port User Name	
	f	
d		

g. Add the connection information for regional1.

Note

For specific information about completing this form, see *Global Manager: Connect to a Regional Manager* in NNMi Help.

h. Click Save and Close in each configuration form to save your changes.

Regional Manager Conr	ection * ¥	
(i) Changes are not co	ommitted until the top-level form is saved!	
-		
Provide the fully-qualified Regional Manager server Connection form for more	i domain name for the hostname of the remote r. See Help \rightarrow Using the Regional Manager a information.	
 Hostname 	regional1.xxxxxxx	
Use Encryption		
 HTTP(S) Port 	80	
 User Name 	system	
User Password	······	
* Ordering	10	

3. Complete steps a through h to establish a connection from globall to regional2.

^{15.} Global Network Management

15.7 Determining the connection status from global1 to regional1 and regional2

To check the connection status from globall to regional1 and regional2, complete the following steps:

1. From the global1 NNMi console, click Global Network Management in the Configuration workspace.

N	etwo	ork Node Manager i
<u>laid</u>	Das	hboards
معن	Incid	dent Management
*	Тор	ology Maps
P	Mon	itoring
▲	Trou	ıbleshooting
-	Inve	ntory
Ω	Man	agement Mode
S	Incid	dent Browsing
%	Integ	gration Module Configuration
æ	Con	figuration
		Communication Configuration
►		Discovery
►		Monitoring
►		Incidents
		Status Configuration
		Global Network Management
•		User Interface
►		Security
►		MIBs
	▦	Device Profiles
•		Object Groups
	▦	RAMS Servers

2. Click the Regional Manager Connections tab.

^{15.} Global Network Management



3. Check the status of regional1 and regional2 by looking at their connection status.

When their connection statuses are shown as **Connected**, it means they are functioning properly. For details, see *Determine the State of the Connection to a Regional Manager* in NNMi Help.

Do not continue to the next section until NNMi completes a good discovery. For details, see 3.3.3 Checking discovery progress.

^{15.} Global Network Management

Do not complete this section until NNMi completes a good discovery. For details, see 3.3.3 Checking discovery progress.

To view the node information regional1 has forwarded to global1, complete the following steps:

1. From the global1 NNMi console, navigate to the Nodes by Management Server form located in the Inventory workspace.

^{15.} Global Network Management

Network Node Manager i							
<u>dil</u>	Dashboards						
م ^ع ہ	Incident Management						
*	Тор	ology Maps					
Ţ	Mon	itoring					
A	Trou	Ibleshooting					
-	Inve	ntory					
	==	Nodes					
	▦	Interfaces					
	▦	IP Addresses					
	▦	SNMP Agents					
	▦	Web Agents					
	▦	IP Subnets					
	▦	VLANs					
	▦	Chassis					
	▦	Cards					
	▦	Ports					
	▦	Node Sensors					
	▦	Physical Sensors					
	▦	Layer 2 Connections					
	▦	Nodes by Management Server					
	▦	Nodes (All Attributes)					
	▦	Interfaces (All Attributes)					
	▦	IP Addresses (All Attributes)					
	▦	MIB Variables					
	▦	Chassis Redundancy Groups					
	▦	Card Redundancy Groups					
	▦	Router Redundancy Groups					
	▦	Router Redundancy Members					

- Hode Groups
- Interface Groups
- 2. Assume that regionall passed information about switch node102130 to globall. After selecting regionall, the inventory might look as follows:

^{15.} Global Network Management

Network Node Manager i		File	⊻iew	Tools	Actions	Help	User N	lame: administrate	or N
Liti	Dashboards								
عه	Incident Management	Nodes b	y Man	agement Serve	er x	-			
4	Topology Maps			5 12 1	% [S	elect	
P	Monitoring		-		C	regional1 -	reg	ionall	M
A	Troubleshooting	Status	Dev	/ Name	▲ Hostna	ame	Manag	ation	De
-	Inventory	0		node100001	node1000	001	10.208.100.1	cm2rack	hita
		0	mode100002 node1		node1000	002	10.208.100.2	cm2desk	hita
		0	111	node1 12018	node1020	018	10.208.102.18	cm2desk	cisc
		0	111	node1 2019	node1020	019	10.208.102.19	cm2desk	ciso
	SNMP Agents	0	TIL			136	10.208.102.36	cm2desk	ala:
	Web Agents	0	One of the critical switches passed from regionall to globall		50	10.208.102.50	cm2desk	hita	
	IP Subnets	\bigcirc			56	10.208.102.66	cm2rack	cisc	
	₩ VLANs	0			98	10.208.102.98	cm2desk	hita	
	III Chassis	o r			14	10.208.102.114	cm2desk	ala:	
	I Cards	0				15	10.208.102.115	cm2desk	ala
	I Ports	0	頭	node102116	node102	116	10.208.102.116	cm2desk	cisc
	H Node Sensors	0	111	node102117	node1021	117	10.208.102.117	cm2desk	ala
	Physical Sensors	0	THE R	node102118	node102	118	10.208.102.118	cm2desk	apr
	HT Laws 2 Consultant								

Repeat steps 1 through 2 to view the device inventory passed to global1 from each other connected regional manager.

^{15.} Global Network Management

15.9 Disconnecting communication between global1 and regional1

To shut down (either temporarily or permanently) a global manager (for example, global1) you must disconnect communication between the global manager and the regional managers.

This example assumes that global1 still has active subscriptions to the regional1 regional manager.

To disconnect communication between global1 and regional1:

1. From the global1 NNMi console, click Global Network Management in the Configuration workspace.

Network Node Manager i							
<u>lad</u>	Das	hboards					
a <mark>بع</mark> ره	Incident Management						
	Тор	ology Maps					
P	Mon	itoring					
A	Trou	ıbleshooting					
10							
Ω	🔬 Management Mode						
8	Inci	dent Browsing					
ବ୍ତ	Inte	gration Module Configuration					
æ	Con	figuration					
		Communication Configuration					
►		Discovery					
►		Monitoring					
►		Incidents					
		Status Configuration					
		Global Network Management					
•		User Interface					
►		Security					
►		MIBs					
	▦	Device Profiles					
►		Object Groups					
	▦	RAMS Servers					

2. Click Regional Manager Connections.



3. Check that the status is **Connected**. If the status is not **Connected**, diagnose the problem using information from the topic *Troubleshoot Global Network Management* in NNMi Help before continuing.

Global Network Management 🗶							
C 🖻 🖻 🗘							
-	Converting Street	Contract Management of	Carlos Andreis Destruction				
For information about this form, click here.	Porwarding Pitter	Regional Manager Connections	Custom Attribute Replication				
	•						
Caution: The system clocks must be synchronized for all NNMi management servers involved in your Global Network Management environment (Regional Managers and Global Managers). For more information, click here.	Optional. Global Managers only: List the Regional Managers from which you want to receive Node object data. For more information, click here.						
	Tip: Consider loggin Node object data.	ng into each Regional Manager and c	onfiguring a Forwarding Filter to	limit the amount of			
	C' 🛊 🗃	C S S 🗎	K 🗲 1-1	of 1 🔶 🕅 🗐			
	Name	Connection State UUID	Des	cription			
	regional1	Connected 306e909c-b5ac-	4523-95c3-291f58223a54 Corp	orate Headquarters			

4. Select regional1, then click the Open icon.



5. Click Connection, select regional1.xx.x. then click the Delete icon.

^{15.} Global Network Management

Global Network Management 3t Regional Manager * X		
C 🖻 🖻 😫 C 📋		
Basics Name Connection State UUID 306e909c-b5ac-4523-95c3-291f58223a54 Description Corporate Headquarters	Configure either HTTP or HTTPS communication with this Regional NNMI manager	nent server. 1 - 1 of 1 -> N 🗊

- 6. Click Save and Close.
- 7. On the **Regional Manager Connections** tab, note the **Name** attribute value for regional1 (case-sensitive). You will need this **Name** attribute in step 9.
- 8. Click Save and Close.
- 9. Type the following command on the command line on global1:

```
nnmnodedelete.ovpl -rm regionall -u NNMiadminUserName -p NNMiadminPassword
```

For -rm, specify the name you noted in step 7.

10. This removes from global1 the node records that regional1 forwarded to it.

The command also closes incidents associated with the nodes forwarded to globall from regional1. For details, see *Disconnect Communication with a Regional Manager* in NNMi Help.

- 11. To remove the configuration records for regional1, do the following.
 - a. Click the Configuration workspace.
 - b. Select the Global Network Management form.
 - c. Click Regional Manager Connections tab.
 - d. Select regional1, then click the Delete icon.

Global Network Management 30								
C 🖻 🖻 🗢								
	(
Facilitation along this form all han	Forwarding Filter	Regional Manager Connections	Custom Attribute Replication					
For information about this form, click here.		•						
Caution: The system clocks must be synchronized for all NNMi management servers involved in your Global Network Management environment (Regional	Optional. Global Managers only: List the Regional Managers from which you want to receive Node object data. For more information, click here.							
Managers and Global Managers). For more information, click here.	Tip: Consider logging into each Regional Manager and configuring a Forwarding Filter to limit the amount of Node object data.							
		_						
	ଟା \star 🗃	C H K 👔	₩ 🗲 1-1of1 → ₩ 🗏					
	Name	Connection State UUID	Description					
	regional1	Connected 306e909c-b5ac-	4523-95c3-291f58223a54 Corporate Headquarters					

e. Click Save and Close to save your deletions.

^{15.} Global Network Management

15.10 Additional information about global network management

15.10.1 Discovery and data synchronization

As network administrators add, delete, and modify devices on a network, the regional servers, such as regional1 and regional2, discover those changes and update the global server, such as global1 in the example in this chapter. regional1 and regional2 also notify global1 of changes that administrators make to the management modes of the nodes they manage.

Note

To maintain consistency, as regional1 and regional2 discover device state changes, they continuously update global1, thereby maintaining identical node states on both the global and regional servers.

Any time global1 requests information about a node that is managed by regional1 or regional2, regional1 or regional2 responds to global1 with the requested information. global1 never talks directly to a node. There will not be duplicate SNMP queries to devices when global1 performs a discovery.

globall synchronizes with regional1 and regional2 each time regional1 or regional2 completes a discovery. NNMi uses forwarding database (FDB) data to calculate Layer 2 connections. FDB data is very dynamic, and varies a lot between discoveries, especially if there are multiple regionals connected to a global server.

Note

Changes to user-modified or application-modified attributes are not updated on the global server during a synchronization.

The **Rediscovery Interval** is adjustable on each regional server, which means there might be differences in the discovery accuracy between globall and the regional managers. The shorter the **Rediscovery Interval**, the more accurate the discovery will be and the more NNMi-generated traffic there will be on the network. The longer the **Rediscovery Interval**, the less accurate the discovery will be and the less NNMi-generated traffic there will be on the network. This means that as your network grows larger, you might want to perform rediscovery less frequently. To set the **Rediscovery Interval**, do the following:

1. From the regional1 or regional2 NNMi console, click **Discovery > Discovery Configuration** in the **Configuration** workspace.



2. Adjust the **Rediscovery Interval** to reflect how often you want the regional server to initiate discovery. The global server will initiate discovery immediately after a regional server completes a discovery.

Discovery Configura	ntion se							
01004	0							
Global Control		Schedule Settings Auto-Discovery Rules Subnet Connection Rules, Unnumbered Ir						
Enable Discovery of								
Interfaces for		Default Interval Setting						
Performance Monitoring		* Rediscovery Interval 24.00 Hours -						
Auto-Discovery Ping Sweep Control (IPv4 only) This control can override the Enable Ping Sweep choice for all Auto- Discovery Brides		Node Group Interval Settings						
		Node Group	5					
Ping Sweep	None 💌	Rediscovery Interval 24.00 Hours 👻						
Sweep Interval 24.00 Hours V Node Name Resolution		Delete Unresponsive Objects Control						
		NNMi deletes objects from the NNMi database after the specified number of unresponsive days. Zero (0) means do NOT delete unresponsive objects. For more information, click here.						
* First Choice	Short DNS Name 💌	Period (in Days) to						
* Second Choice	Short sysName	Delete Unresponsive D Nodes						
Third Choice	IP Address 💌	Period (in Days) to Pelete Connections 0						

3. Click Save and Close.

15.10.2 Replicating custom attributes from a regional manager to the global manager

NNMi enables you to set custom attributes on a regional manager and replicate those custom attributes to the global manager. For example, you can add custom attribute data to nodes on a regional manager and, after replicating that data to the global manager, use that data to enrich incidents for those nodes.

Note

NNMi supports replication of custom attributes from a regional manager to a global manager for nodes and interfaces.

You can configure custom attribute replication on the NNMi console using the global manager's **Custom Attribute Replication** tab (within the **Global Network Management** configuration).



NOTE

NNMi replicates custom attributes for unnumbered interfaces without any user configuration or input. For details, see NNMi Help.

15.10.3 Status polling or configuration polling a device

This example assumes the following:

- Regional NNMi management server regional2 discovers and manages Node X.
- Global NNMi management server global1 connects with regional NNMi management server regional2.

See the following figure.

^{15.} Global Network Management

Figure 15-2: Status polling or configuration polling a node



To status poll Node X from global1, do the following:

- 1. From global1, click Nodes in the Inventory workspace.
- 2. Select Node X from the nodes inventory.
- 3. Use the Actions > Polling > Status Poll menu to request a status poll of Node X.
- 4. NNMi management server globall requests a status poll from regional NNMi management server regional2 and shows the results on your screen.

Initiate the status poll request from global1 or regional2.

You will see the same status poll results whether you initiate the status poll request from globall or regional2.

If you want global1 to have the most current discovery information for Node X, do the following to configuration poll Node X from global1:

- 1. From global1, click Nodes in the Inventory workspace.
- 2. Select Node X from the nodes inventory.
- 3. Use the Action > Polling > Configuration Poll menu to request a configuration poll of Node X.
- 4. NNMi management server globall requests a configuration poll from regional NNMi management server regional2 and shows the results on your screen.

You will see the same configuration poll results whether you initiate the configuration poll request from globall or regional2.

15. Global Network Management

15.10.4 Determining device status and NNMi incident generation using a global manager

NNMi management server globall monitors for state changes arriving from regional managers regionall and regional2 and updates the states in its local database.

The NNMi State Poller service on each NNMi management server (regional1 and regional2) calculates state values for the devices it monitors. global1 receives state value updates from regional1 and regional2. global1 polls nodes that it discovers, and does not poll nodes being managed by regional1 and regional2.

When you change the management mode of a node being managed by regional1, you will see that management mode change on global1 as well. As network administrators add, remove, and modify network equipment being managed by regional1 or regional2, regional1 or regional2 updates global1 of these network device changes.

globall generates incidents using its own causal engine and topology, including the node object data forwarded to it by regionall and regional2. This means that the incidents it generates might be slightly different from the regional1 and regional2 incidents if there are differences in topology.

It is better to avoid using a forwarding filter on regional1 and regional2, as filtering might affect the connectivity on global1. The result could be a difference in the root cause analysis between global1 and the two regionals (regional1 and regional2). In most cases, if you choose to avoid using forwarding filters, a global NNMi management server will have a larger topology. This helps it draw more accurate root cause analysis conclusions.

Without additional configuration, regionall does not forward traps to globall. To do this, you must configure regional1 to forward specific traps to global1. We recommend that you configure regional managers to forward only low-volume, important traps so as to avoid excessive burden on the global manager. NNMi drops forwarded traps when the forwarded traps result in a TrapStorm incident. See the TrapStorm Management Event details in the NNMi console.

^{15.} Global Network Management
15.11 Troubleshooting tips for global network management

15.11.1 Troubleshooting information in NNMi Help

For global network management troubleshooting information, see *Troubleshoot Global Network Management* in NNMi Help.

15.11.2 Clock synchronization

All NNMi management servers in your network environment that participate in global network management (global managers and regional managers) or single sign-on (SSO) must have their internal time clocks synchronized in universal time. Use a time synchronization program, such as the Linux Network Time Protocol Daemon (NTPD) tool or one of the available Windows operating system tools.

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

```
\ensuremath{\texttt{NNMi's}} self monitoring has detected a problem (Minor). Please see Help > System Infor mation > Health for details.
```

Check the nnm.log file on the global manager for the following message:

```
SEVERE
[com.hp.ov.nms.topo.spi.server.bridge.BridgeConnectionSelectorImpl] Not connecting to
system server-name due to clock difference of number-of-seconds. Remote time is date
-time.
```

Perhaps the clocks have drifted apart and need to be resynchronized.

NNMi will disconnect the regional manager connection within a few minutes of output of this message to the log.

NNMi Self Monitoring will also detect the following problem:

[Minor] The connection to the regional manager 'name' is down.

15.11.3 Global network management system information

To view information about your global network management connections, choose **Help**, **System Information**, and then click the **Global Network Management** tab.

15.11.4 Synchronizing regional manager discovery from a global manager

If you notice an information inconsistency between global1 and regional2, run the nnmnoderediscover.ovpl script from global1, which will cause global1 and regional2 to synchronize. This also results in regional2 updating global1 with any new discovery results. This example uses the network shown in the following figure.

```
15. Global Network Management
```

Figure 15-3: Global network management



Run the following command to synchronize nodes X, Y, and Z with global1:

nnmnoderediscover.ovpl -u username -p password -rm regional2

For details, see the nnmnoderediscover.ovpl Reference Page.

Note the following:

• NNMi automatically resynchronizes topology, state, and status following a manual resynchronization.

15.12 Upgrading NNMi in a global network management environment

To upgrade NNMi management servers configured in a global network management environment, see 24.3 Upgrading global and regional managers from NNMi 12-10.

^{15.} Global Network Management

15.13 Global network management and address translation protocol

Each group of dynamic network address translation (NAT) or dynamic port address translation (PAT) or dynamic network address and port translation (NAPT) requires an NNMi regional manager, in addition to a tenant that is unique within the entire NNMi global network management configuration. For details, see 13. Managing Overlapping IP Addresses in a NAT Environment and NNMi Help.

^{15.} Global Network Management

16

NNMi IPv6 Management Feature

You must purchase and install an NNMi Advanced license to use the IPv6 management feature. References to NNMi in this chapter assume an NNMi with an NNMi Advanced license installed. IPv6 management in NNMi enables the discovery and monitoring of IPv6 addresses, including their interfaces, nodes, and subnets. To provide a seamless integration, NNMi extends its IP Address model to include both IPv4 and IPv6 addresses. Whenever possible, NNMi treats all IP addresses equally; most of the features associated with an IPv4 address are also available for IPv6 addresses. However, there are some exceptions. For details about IPv6 information displayed on the NNMi console, see NNMi Help.

16.1 Overview of the NNMi IPv6 management feature

The NNMi IPv6 management feature provides the following:

- IPv6 inventory discovery for IPv6-only and dual-stacked devices
 - IPv6 addresses
 - IPv6 subnets
 - Associations between IPv6 addresses, subnets, interfaces, and nodes
- Native IPv6 SNMP communication for the following:
 - Node discovery
 - Interface monitoring
 - Trap and inform reception and forwarding
- Automatic selection of IPv4 or IPv6 communication (management addresses) for dual-stacked devices On the NNMi console, use **Communication Configuration** in the **Configuration** workspace to set the SNMP management address preference to IPv4 or IPv6.
- Native ICMPv6 communication for IPv6 Address fault monitoring
- Seeded device discovery using an IPv6 address or host name
- Automatic IPv6 device discovery using IPv6 Layer 3 neighbor discovery hints
- Automatic IPv6 device discovery using Layer 2 neighbor discovery hints using Link Layer Discovery Protocol (LLDP) IPv6 neighbor information
- Consolidated presentation of IPv4 and IPv6 information
 - Inventory views for nodes, interfaces, addresses, subnets, and associations
 - Layer 2 Neighbor View and Topology Maps for IPv4 and IPv6 devices
 - Layer 3 Neighbor View and Topology Maps for IPv4 and IPv6 devices
 - Incidents, conclusions, root-cause analysis
- NNMi console actions: ping and traceroute for IPv6 addresses and nodes
- NNMi configuration using IPv6 addresses and address ranges
 - Communication configuration
 - Discovery configuration
 - Monitoring configuration
 - Node & Interface Groups
 - Incident configuration
- DTK Web-services support for IPv6 inventory and incidents

The NNMi IPv6 management feature excludes the following:

- Use of IPv6 Ping sweep for discovery
- IPv6 Network Path View (Smart Path)
- IPv6 Link Local Address fault monitoring
- Using IPv6 Link Local Addresses as discovery seeds

16.2 Prerequisites for using the NNMi IPv6 management feature

For details about the management server specifications and about NNMi installation, see the Release Notes.

To use native IPv6 communication, the NNMi management server must be a dual-stacked system, meaning that it communicates using both IPv4 and IPv6.

Additional requirements for IPv6 include the following:

- You must enable and configure IPv4 on at least one network interface.
- You must enable IPv6 and have a unicast address (that is not a link-local unicast address), such as a global unicast address or a unique local unicast address, configured on at least one network interface that is connected to the IPv6 network you must manage.
- You must configure IPv6 routes on the NNMi management server to enable NNMi to communicate with any devices you want NNMi to discover and monitor using IPv6.

Note

You can use an IPv4-only NNMi management server, but doing so will limit NNMi from fully managing IPv4/IPv6 dual-stacked devices. For example, if you use an IPv4-only management server, NNMi cannot discover IPv6-only devices, cannot discover using IPv6 seeds and hints, and cannot monitor for faults on devices with IPv6 addresses.

The DNS server used by the NNMi management server must resolve host names to and from IPv6 addresses. That means the DNS server must map a host name to a 128-bit IPv6 address. If an IPv6-capable DNS server is not available, NNMi will still function correctly; however, NNMi does not determine or display DNS host names for nodes using IPv6 addresses.

^{16.} NNMi IPv6 Management Feature

As mentioned earlier, you must purchase and install an NNMi Advanced license to use the IPv6 management feature. For details about obtaining and installing an NNMi Advanced license, see 2. Installing and Uninstalling NNMi.

The NNMi product includes a temporary Instant-On license password. This is a temporary, but valid NNMi Advanced license. Make sure to obtain and install a permanent license password as soon as possible.

^{16.} NNMi IPv6 Management Feature

16.4 Environment supported by the NNMi IPv6 management feature

For details about the supported operating system configurations for NNMi, see the Release Notes.

16.4.1 Types of NNMi management servers and supported functions

The following table shows the capabilities of both IPv4-only and dual-stacked NNMi management servers.

Table 16-1: Management server capabilities

Feature/capability	IPv4-only	Dual-stack
IPv4 Communication (SNMP, ICMP)	Supported	Supported
IPv6 Communication (SNMP, ICMPv6)	Not supported	Supported
Dual-Stack Managed Node	Supported	Supported
Discovery using IPv4 Seed	Supported	Supported
Discovery using IPv6 Seed	Not supported	Supported
IPv4 Address and Subnet Inventory	Supported	Supported
IPv6 Address and Subnet Inventory	Supported	Supported
Interface Status and Performance using SNMP	Supported	Supported
IPv4 Address Status using ICMP	Supported	Supported
IPv6 Address Status using ICMPv6	Not supported	Supported
IPv6-only Managed Node	Not supported	Supported
IPv4-only Managed Node	Supported	Supported

16.4.2 Supported SNMP MIBs for IPv6

NNMi supports the following SNMP MIBs for IPv6:

- RFC 4293 (current IETF standard)
- RFC 2465 (original IETF proposal)
- Cisco IP-MIB

^{16.} NNMi IPv6 Management Feature

During NNMi installation, the installation script activates the IPv6 feature; however, you can deactivate this IPv6 feature manually, if desired, by editing the nms-jboss.properties file.

You can also reactivate the IPv6 feature after it has been deactivated. For details, see 16.6 Deactivating the IPv6 management feature and 16.7 Reactivating the IPv6 management feature.

16. NNMi IPv6 Management Feature

16.6 Deactivating the IPv6 management feature

You can administratively disable the IPv6 capabilities using either of the following methods:

- 1. Turn off the IPv6 master switch in the nms-jboss.properties file, then restart NNMi.
- 2. Let the NNMi Advanced license expire, or replace it with a basic NNMi license.

To administratively disable the IPv6 feature by using method 1:

1. Open the nms-jboss.properties file.

Locate the following line:

- Windows: %NNM_PROPS%\nms-jboss.properties
- Linux: \$NNM_PROPS/nms-jboss.properties

NNMi provides a complete description of each property, showing them as comments in the nms-jboss.properties file.

2. To deactivate IPv6 communication in NNMi:

a. Locate the text that begins with # Enable Java IPv6 Communication.

b. Locate the following line:

java.net.preferIPv4Stack=false

c. Edit the line to read as follows:

java.net.preferIPv4Stack=true

Make sure the line is not commented.

- 3. To deactivate overall IPv6 management in NNMi:
 - a. Locate the text that begins with # Enable NNMi IPv6 Management.
 - **b.** Locate the following line:

com.hp.nnm.enableIPv6Mgmt=true

c. Edit the line to read as follows:

com.hp.nnm.enableIPv6Mgmt=false

Make sure the line is not commented.

d. Save and close the nms-jboss.properties file.

4. Execute the following commands to restart NNMi:

```
ovstop
ovstart
```

Important

If you change a file while the system is operating in High Availability (HA) mode, you must make the change on both nodes in the cluster. For NNMi in an HA configuration, if the change requires you to stop and restart the NNMi management server, you must place the nodes in maintenance mode before running the ovstop and ovstart commands. For details, see 19.6.1 Placing NNMi in maintenance mode.

5. Check the NNMi processes by using the following command:

ovstatus -v ovjboss

The following subsections describe NNMi behavior and inventory cleanup after you disable IPv6.

16.6.1 IPv6 monitoring following deactivation of the IPv6 management feature

If IPv6 management or IPv6 communication becomes completely disabled, the State Poller service immediately stops monitoring IPv6 addresses with ICMPv6. NNMi sets the IP address state of these addresses to **Not Polled**. If you select such an address and then use the **Actions** > **Configuration Details** > **Monitoring Settings** command for that address, NNMi will display Fault ICMP Polling enabled: false even though the associated **Monitoring Configuration** rule has enabled **IP Address Fault Polling**.

16.6.2 IPv6 inventory following deactivation of the IPv6 management feature

Once NNMi completely discovers your IPv6 inventory, you can enable NNMi to clean it up automatically in the following scenarios:

You turned on the master IPv6 switch, then turned it off and restarted NNMi.
 NNMi does not immediately remove the IPv6 inventory. NNMi removes the IPv6 inventory for SNMP nodes during the next discovery cycle. For nodes whose management address is an IPv6 address, their management addresses remain as is. NNMi does not remove non-SNMP IPv6 nodes. You must manually delete the nodes whose IPv6 data remains from the NNMi inventory.

16.6.3 Known issues when cleaning Up IPv6 inventory

You could experience leftover IPv6 inventory in the following situation: Suppose that NNMi successfully uses SNMP to manage an IPv6 node, then the node becomes inaccessible before the next discovery. Due to the design of the existing discovery system, the discovery process cannot update a node that loses its ability to communicate using SNMP. To remove such remaining nodes, first fix the communication problem, then, from the NNMi console, on the **Actions** menu choose **Polling**, and then the **Configuration Poll** command to obtain configuration information from these nodes. For a native IPv6 node, delete the node directly from the NNMi console.

16.7 Reactivating the IPv6 management feature

Capabilities requiring IPv6 communication, such as the discovery of only IPv6 devices and monitoring IPv6 address status, require an NNMi management server to have an IPv6 global unicast address configured and operational.

To reactive the IPv6 feature after it has been deactivated:

1. Edit the nms-jboss.properties file.

Locate the following line:

- Windows: %NNM_PROPS%\nms-jboss.properties
- Linux: \$NNM_PROPS/nms-jboss.properties

NNMi provides a complete description of each property, showing them as comments in the nmsjboss.properties file.

- 2. Locate the text that begins with # Enable NNMi IPv6 Management.
- 3. To enable IPv6 communication in NNMi, un-comment the following property:

java.net.preferIPv4Stack=false

To un-comment a property, remove the #! characters from the beginning of the line.

- 4. Locate the text that begins with # Enable NNMi IPv6 Management.
- 5. To enable overall IPv6 management in NNMi, un-comment the following property:

com.hp.nnm.enableIPv6Mgmt=true

- 6. Save and close the nms-jboss.properties file.
- 7. Execute the following commands to restart NNMi:

```
ovstop
ovstart
```

Important

If you change a file while the system is operating in High Availability (HA) mode, you must make the change on both nodes in the cluster. For NNMi in an HA configuration, if the change requires you to stop and restart the NNMi management server, you must place the nodes in maintenance mode before running the ovstop and ovstart commands. For details, see 19.6.1 Placing NNMi in maintenance mode.

8. Check the NNMi processes by using the following command:

```
ovstatus -v ovjboss
```

Successful startup will look something like the following:

```
object manager name: ovjboss
state: RUNNING
PID: process-ID-number
last message: Initialization complete.
exit status: -
additional info:
SERVICE STATUS
CommunicationModelService Service is started
CommunicationParametersStatsService Service is started
```

CustomPoller Service is started IslandSpotterService Service is started ManagedNodeLicenseManager Service is started MonitoringSettingsService Service is started NamedPoll Service is started msApa Service is started NmsCustomCorrelation Service is started NmsDisco Service is started NmsEvents Service is started NmsEventsConfiguration Service is started NmsExtensionNotificationService Service is started NnmTrapService Service is started PerformanceSpiAdapterTopologyChangeService Service is started PerformanceSpiConsumptionManager Service is started RbaManager Service is started RediscoverQueue Service is started SpmdjbossStart Service is started StagedIcmp Service is started StagedSnmp Service is started StatePoller Service is started TrapConfigurationService Service is started TrustManager Service is started

- After you reactivate IPv6, NNMi views immediately include the IPv6 inventory for newly discovered nodes.
 During the next discovery cycle, NNMi views will show the IPv6 inventory associated with previously discovered nodes.
- 10. Optionally, you can set the SNMP management address preference for dual-stacked managed nodes.

Dual-stacked managed nodes are nodes that can communicate using either IPv4 or IPv6. To do this, complete the following steps:

a. From the NNMi console, click Communication Configuration located in the Configuration workspace.

b. Locate the **Management Address Selection** section. Select IPv4, IPv6, or Any in the **IP Version Preference** field.

c. Save your changes.

d. Execute the following commands to restart NNMi:

```
ovstop
ovstart
```

Important

If you change a file while the system is operating in High Availability (HA) mode, you must make the change on both nodes in the cluster. For NNMi in an HA configuration, if the change requires you to stop and restart the NNMi management server, you must put the nodes in maintenance mode before running the ovstop and ovstart commands. For details, see 19.6.1 Placing NNMi in maintenance mode.

To speed things up, select nodes that you know are dual-stacked nodes, and then use the Actions > Configuration Poll command located on the NNMi console. You can also use the nnmnoderediscover.ovpl script to add nodes to the NNMi discovery queue. For details, see the *nnmnoderediscover.ovpl Reference Page*.

After you enable IPv6 communication on the NNMi management server, NNMi begins using ICMPv6 to monitor nodes for IPv6 address faults.

^{16.} NNMi IPv6 Management Feature



NNMi Data Resilience

This chapter describes how NNMi protects the NNMi data in case of hardware failure.

17.1 Approaches to NNMi data resilience

NNMi supports two approaches to protecting the NNMi data in case of hardware failure:

• Application failover

NNMi application failover provides for disaster recovery by maintaining a copy of the NNMi database transaction logs on an identically configured system. For details, see 18. Configuring NNMi for Application Failover.

• Running in a high availability (HA) cluster

Running NNMi in an HA cluster provides for nearly 100 percent availability of the NNMi management server by maintaining the NNMi database and configuration files on a shared disk. For details, see 19. Configuring NNMi in a High Availability Cluster.

In both approaches, if the current NNMi management server fails, the second system automatically becomes the NNMi management server.

17.2 Comparison of approaches to NNMi data resilience

The following table compares several aspects of these two approaches to NNMi data resilience.

Comparison item	NNMi application failover	NNMi running in an HA cluster	
Required software products	NNMi	NNMiA separately purchased HA product	
Time to failover	Time needed to process the transaction logs (under normal conditions, 10 to 60 minutes)	Under normal conditions, 5 to 30 minutes	
Transparency of failover	Partial. The IP address of the NNMi management server changes to the physical address of what was the standby server. Users must use the new IP address to connect to the NNMi console.	Complete. All connections use the virtual IP address of the HA cluster, which does not change in the event of failover.	
Relative proximity of active and standby servers	LAN or WAN	LAN or WAN (some HA products only)	
Interaction with global network management	 Application failover Global and regional managers cannot be configured for application failover. HA Each global manager and each regional manager can be configured for HA. Each of these configurations requires two physical systems. When failover occurs on a global manager or regional manager, NNMi re-establishes the connections between the global managers and regional managers. 		
NNMi maintenance	NNMi must be taken out of the application failover cluster before applying a patch or upgrading.	NNMi can be patched and upgraded without unconfiguring HA.	

Table 17-1:	NNMi data	resilience	comparison
		1001100	eempaneen

18

Configuring NNMi for Application Failover

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

18.1 Application failover overview

The application failover feature multiplexes the NNMi management server without using a shared disk or cluster software.

When you configure two NNMi management servers, one as the active server and the other as a standby server, you can continue monitoring the network by transferring the NNMi functionality to the standby server when an error occurs in the active server.

This application failover feature is implemented based on NNMi's unique cluster manager (nnmcluster process) control and has features that differ from those of an HA configuration, which links to cluster software. Note that the application failover configuration is sometimes referred to as the *NNMi cluster* (or simply the *cluster*) in the manual and in Help.



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

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

- Active: The server that is running network monitoring.
- Standby: The server in the NNMi cluster that is waiting for a failover event; this server is not running network monitoring.
- Cluster member: A Java process running on a system that is using JGroups technology to connect to a cluster; you can have multiple members on a single system.
- Postgres: The database NNMi uses to store information such as topology, incidents, and configuration information.
- Cluster manager: The nnmcluster process and tool used to monitor and manage the servers for the application failover feature.

^{18.} Configuring NNMi for Application Failover

18.2 Application failover basic setup

To deploy the application failover feature, install NNMi on two servers. These two NNMi management servers are referred to in this chapter as the *active* server and the *standby* server. During normal operation, only the active server is running network monitoring.

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

You can use either of the following methods to configure the application failover feature:

- Configuring application failover manually
- Configuring application failover with the NNMi Cluster Setup Wizard

18.2.1 Prerequisites for setting up application failover

For application failover to work successfully, the NNMi management servers must meet the following requirements:

• Only a configuration in which NNMi is used alone is supported.

A configurations in which NNMi is linked to another related program product, such as JP1, is not supported. In such a case, use an HA configuration using cluster software.

- It must be possible to resolve the host name and IP address of the active server and of the standby server at both NNMi management servers.
- On the network, all IPv4 addresses of both NNMi management servers must not overlap.
- Both NNMi management servers must be running the same operating system. For example, if the active server's operating system is Microsoft^(R) Windows Server^(R) 2012 R2 Datacenter, the standby server's operating system must also be Microsoft^(R) Windows Server^(R) 2012 R2 Datacenter.
- Both NNMi management servers must be running the same NNMi version. The NNMi patch levels must also be the same on both servers. For example, if NNMi 12-00 is running on the active server, the identical NNMi version, NNMi 12-00, must be on the standby server.
- The system password must be the same on both NNMi management servers.
- Do not completely disable HTTP access to NNMi before configuring application failover. For details, see 21.18 Configuring NNMi to require encryption for remote access. After successfully configuring the application failover cluster, you can disable HTTP and other unencrypted access.
- For an installation on Windows, NNMi must be installed in the same directory in both NNMi management servers and the %NnmDataDir% and %NnmInstallDir% system variables must be set to identical values on both servers.
- Both NNMi management servers must have identical licensing attributes (number of nodes managed, and whether NNMi or NNMi Advanced is used). For example, the node counts and licensed features must be identical.

Important

The standby server must also have the identical license.

• Do not enable application failover until NNMi is in an advanced stage of initial discovery. For details, see 6.4 Evaluating discovery.

^{18.} Configuring NNMi for Application Failover

- For application failover to function correctly, the active and standby servers must have unrestricted network access to each other. Any software that locks files or restricts network access can cause NNMi communication problems. Configure these applications to ignore the files and ports used by NNMi.
- The active server and the standby server must be able use all of the IPv4 addresses of the NIC used for cluster communication to communicate with the other NNMi management server for which application failover is configured. If there is any IPv4 address that cannot be used to communicate with the other management server, application failover configuration might fail, or the behavior of a configured application failover environment might become corrupted. For details on how to specify the NICs to be used for cluster communication, see 18.3.3 Setting application failover communications.
- We do not recommend that you set up a firewall between the active and standby servers. When setting up a firewall, configure the systems such that both servers can use all ports for communication.
- To run a firewall in an NNMi management server, permit communication between processes in the local server and communication with remote servers at all ports. Application failover performs communication dynamically using any available port.
 - If you are using a firewall that permits communication on a process-by-process basis (such as Windows Firewall), permit communication by the cluster manager (nnmcluster.exe).
 - If you are using a firewall that permits communication on a port-by-port basis, permit the following types of communication:

IP addresses: All IP addresses assigned to the local server and remote servers Ports: All ports

• The same password must be set for the NNMi databases on both the active and standby servers.

If you have changed the password for an NNMi database, set the same password on all servers before you start configuring application failover.

If all these conditions are satisfied, complete the steps shown in 18.3 Configuring NNMi for application failover. For details, see E. List of Ports Used by NNMi.

18.2.2 Notes on application failover

This subsection provides additional useful information about application failover.

• In the application failover configuration, NNMi fails over when its server stops, but does not fail over when an NNMi process stops (except for nnmcluster). For details, see 18.4.2 Application failover scenarios.

If you want NNMi to fail over when an NNMi process stops, use an HA configuration that is based on HA cluster software.

- There is no functionality for providing a report when the standby server stops (the state in which no switchover target is available).
- There is no functionality for executing user-specified commands for performing some processing when failover occurs.
- The IP address of the server on which NNMi runs switches when failover occurs. The IP address is not inherited. Therefore, note the following points:
 - Set up the SNMP trap transmission target in both NNMi management servers.
 - Register the bookmarks of both NNMi management servers in the Web browser, and connect it to the active server.

^{18.} Configuring NNMi for Application Failover

- Back up data periodically so that you will be able to use the backed up data in the unlikely event of a system failure. The data duplicated in the standby server by the application failover feature cannot be used as a substitute for a backup.
- In the application failover configuration, the database requires three times the amount of disk space that would be needed in an ordinary configuration. For details about database configuration, see 18.4.1 Application failover behavior.

^{18.} Configuring NNMi for Application Failover

This section explains how to configure NNMi for application failover.

18.3.1 Configuring application failover manually

To configure NNMi for application failover:

1. Install NNMi on the active server (server X) and the standby server (server Y).



- 2. Install a permanent license in each server, as described in 2.3 Licensing NNMi.
- 3. Run the ovstop command on each server to shut down NNMi.
- 4. Configure server X (active) and server Y (standby) for the application failover feature using guidance from the detailed instructions contained in the nms-cluster.properties file.

Use the procedure shown below. *Edit* in the following steps means to uncomment (remove the leading # ! from) the lines in the text block within the file and to modify the text.

a. Edit the following file:

```
Windows:
%NnmDataDir%shared\nnm\conf\props\nms-cluster.properties
Linux:
```

\$NnmDataDir/shared/nnm/conf/props/nms-cluster.properties

b. Declare a unique name for the NNMi cluster. Use the same name when configuring both the active and standby servers. Specify alphanumeric characters for the name, which is case-sensitive.

Specifying this parameter enables the application failover feature:

com.hp.ov.nms.cluster.name=MyCluster

c. Add the host names of all nodes in the cluster to the com.hp.ov.nms.cluster.member.hostnames parameter in the nms-cluster.properties file:

```
com.hp.ov.nms.cluster.member.hostnames = fqdn_for_active,fqdn_for_standby
```

Important

If either of the nodes has multiple IPv4 addresses, enter the IPv4 address to be used for each node's NNMi communication in the com.hp.ov.nms.cluster.member.hostnames parameter.

5. Configure the NNMi certificate (use nnm-key.p12 and nnm-trust.p12 files or a certification organization).

Depending on the method you selected, complete the set of instructions described in 10.3.5 Working with Certificates in Application Failover Environments.

Important

When configuring the application failover feature, you must merge the nnm-trust.p12 file content for both nodes into a single nnm-trust.p12 file. You must choose your approach and complete one set of instructions.

6. Back up the source files, which will be used when canceling the application failover configuration, and then copy the following file from server X to server Y:

Windows:

```
%NnmDataDir%shared\nnm\conf\nnmcluster\cluster.keystore
```

Linux:

```
$NnmDataDir/shared/nnm/conf/nnmcluster/cluster.keystore
```

7. Configure the NIC that is used for cluster communication between both nodes.

For details, see 18.3.3 Setting application failover communications.

Important

Make sure to specify the setting if either of the nodes has multiple NICs.

8. Run the following command on both server X and server Y:

nnmcluster

Each server displays something similar to the following:

Important

To terminate the command, press the Enter key and then enter the command quit.

If NNMi is configured correctly, the display lists both server X and server Y. If information about both nodes is not displayed, the nodes are not communicating with each other. Here are some things to check for and correct before continuing:

- Specify the same cluster name in com.hp.ov.nms.cluster.name on both servers, as follows:
 - Windows: %NnmDataDir%shared\nnm\conf\props\nms-cluster.properties
 - Linux: \$NnmDataDir/shared/nnm/conf/props/nms-cluster.properties

• The key CRCs might be different on server X and server Y.

Check the contents of the following file on both server X and server Y:

- Windows:

```
%NnmDataDir%shared\nnm\conf\nnmcluster\cluster.keystore
```

- Linux:

```
$NnmDataDir/shared/nnm/conf/nnmcluster/cluster.keystore
```

If the key CRCs are different, perform step 6.

- A firewall on server X or server Y might be preventing the nodes from communicating. Configure the nodes so that they can communicate.
- Make sure you merged the nnm-trust.p12 files. You should see this error displayed after running the nnmcluster command.
- Make sure that the IP address that is obtained from the NIC specified in com.hp.ov.nms.cluster.interface matches the IP address that will be resolved from the host name specified in com.hp.ov.nms.cluster.member.hostnames.

Specify com.hp.ov.nms.cluster.interface in the following file:

- Windows: %NnmDataDir%Conf\nnm\props\nms-cluster-local.properties

- Linux: \$NnmDataDir/conf/nnm/props/nms-cluster-local.properties

Specify com.hp.ov.nms.cluster.member.hostnames in the following file:

- Windows: %NnmDataDir%shared\nnm\conf\props\nms-cluster.properties
- Linux: \$NnmDataDir/shared/nnm/conf/props/nms-cluster.properties
- Make sure that the remote server's standby IP address matches the IP address specified for the remote server. Check that the IP address available to the remote server for cluster communication matches the IP address specified for the remote server for cluster communication.

The default for the port used for cluster communication is 7800.

You can use the netstat command to check the standby IP address.

To determine the IP address specified for the remote server for cluster communication, check the com.hp.ov.nms.cluster.member.hostnames parameter in the nms-cluster.properties file.

If a host name is specified in com.hp.ov.nms.cluster.member.hostnames, check the IP address that can be resolved from the host name.

• Server X and server Y are running different operating systems.

For example, server X might be running a Linux operating system and server Y a Windows operating system. Specify the configuration in an environment in which the same operating system is running.

• Server X and server Y are running different NNMi versions.

For example, server X might be running NNMi 11-00 and server Y a patched version of NNMi 11-00. Specify the configuration in an environment in which the same version of NNMi is installed.

9. On server X, start the NNMi cluster manager:

```
nnmcluster -daemon
```

After you run the nnmcluster -daemon command on NNMi management server X, the NNMi cluster manager goes through the following startup routine:

- Connects NNMi management server X to the cluster.
- Detects that there are no other NNMi management servers present.

- NNMi management server X assumes the active state.
- Starts the NNMi services on NNMi management server X (the active server).
- Creates a database backup.

For details, see the nnmcluster Reference Page.

- 10. Wait a few minutes for server X to become the first active server in the cluster. Run the nnmcluster -display command on server X and search the displayed results for the term ACTIVE, such as ACTIVE_NNM_STARTING or ACTIVE_SomeOtherState. Do not continue with step 11 until you have confirmed that server X is the active server.
- 11. On server Y, start the NNMi cluster manager:

```
nnmcluster -daemon
```

After you run the nnmcluster -daemon command on NNMi management server Y, the NNMi cluster manager goes through the following startup routine:

- Connects NNMi management server Y to the cluster.
- Detects that NNMi management server X is present and is in the active state. The display shows STANDBY INITIALIZING.
- Compares the database backup on NNMi management server Y to the backup on NNMi management server X. If these do not match, a new database backup is sent from NNMi management server X (active) to NNMi management server Y (standby). The display shows STANDBY_RECV_DBZIP.
- NNMi management server Y receives a minimal set of transaction logs, which is the minimum necessary for the backup to be applicable for its standby state. The display shows STANDBY_RECV_TXLOGS.
- NNMi management server Y goes into a waiting state, continuously receiving new transaction logs and heartbeat signals from NNMi management server X. The display shows STANDBY READY.

For details, see the nnmcluster Reference Page.

12. If a failover occurs, the NNMi console for server X no longer functions. Close the NNMi console session for server X and log on to server Y (the new active server).

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

13. Change the configuration of the devices being monitored by NNMi to send traps to both server X and server Y. While server X (active) is running, it processes the forwarded traps and server Y (standby) ignores the forwarded traps.

18.3.2 Configuring application failover with the NNMi Cluster Setup Wizard

The NNMi Cluster Setup Wizard automates the process of configuring a cluster within NNMi for use with application failover. The wizard enables you to do the following:

- Specify and validate cluster nodes
- Define cluster properties and ports
- Merge the nnm-key.p12 file and nnm-trust.p12 file content for both nodes into a single nnm-key.p12 file and nnm-trust.p12 file

^{18.} Configuring NNMi for Application Failover

1. Launch the Cluster Setup Wizard by entering the following URL into a supported Web browser:

http://NNMi-server:port/cluster

- *NNMi-server* is the value of the NNMi host.
- *port* is the value of the NNMi port.
- 2. Enter your system User Name and Password, and then click the Login button to sign into NNMi.
- 3. Enter Local Hostname and Remote Cluster Node values to define the cluster nodes, and then click Next.

Important

If either of the nodes has multiple IPv4 addresses, enter the IPv4 address to be used for each node's NNMi communication in **Local Hostname** and **Remote Cluster Node**.

4. On the **Communication Results** page, review the communication verification results. If an error is detected, click **Previous** and fix the problem; otherwise, click **Next**.

A green status message indicates that connection to the remote cluster node was successful.

- 5. On the **Define Cluster Properties** page, enter the **Cluster Name** and define the **Backup Interval (in hours)**. Specify **Cluster Name** in alphanumeric characters. Next, specify whether to enable automatic failover. Click **Next**.
- 6. On the **Define Cluster Ports** page, enter **Starting Cluster Port** and **File Transfer Port** values. The NNMi cluster uses four contiguous ports beginning with the **Starting Cluster Port**.
- 7. Click Next.
- 8. Review the summary information that was entered on the **Summary** page.

Click **Previous** to go back and change configuration information; otherwise, click **Commit** to save the cluster configuration.

The final summary indicates that the information was written successfully to the configuration files.

- 9. Immediately stop NNMi on both nodes by running the ovstop command on both nodes.
- 10. Configure the NIC that will be used for cluster communication between the nodes.

For details, see 18.3.3 Setting application failover communications.

Important

Make sure to specify the setting if either of the nodes has multiple NICs.

- 11. Verify that the two nodes are able to cluster by running the nnmcluster command on both nodes. If the nodes are not able to cluster, see 18.3 Configuring NNMi for application failover.
- 12. Start NNMi on the desired active node with the nnmcluster -daemon command. Wait for NNMi to report ACTIVE.

For details, see 18.3 Configuring NNMi for application failover.

13. Start the standby node with the nnmcluster -daemon command.

18.3.3 Setting application failover communications

During installation, NNMi queries all network interface cards (NICs) on the system to find one to use for cluster communications. If your system has multiple NICs, you can use the following procedure to choose the NIC to use for nnmcluster operations:

^{18.} Configuring NNMi for Application Failover

Important

Make sure to specify the setting if either of the nodes has multiple NICs.

1. Run nnmcluster -interfaces to list all available interfaces.

For details, see the *nnmcluster Reference Page*.

- 2. Edit the following file:
 - Windows: %NnmDataDir%conf\nnm\props\nms-cluster-local.properties
 - Linux: \$NnmDataDir/conf/nnm/props/nms-cluster-local.properties

3. Locate the line containing text similar to the following:

com.hp.ov.nms.cluster.interface=value

4. Change the value as desired.

The interface value must pertain to a valid interface; if it does not, the cluster might not start.

An example of the value is eth3 that was output by nnmcluster -interfaces in step 1.

In Windows, the description of the system interface is displayed following the value such as eth3.

Map the interface to be used with the value such as eth3 by checking the description of the interface by using a command such as ipconfig /all.

In Linux, the name of the interface is displayed. Check the name of the interface to be used by using a command such as ifconfig.

Important

The active server and the standby server must be able use all of the IPv4 addresses of the NIC used for cluster communication to communicate with the other NNMi management server for which application failover is configured. If there is any IPv4 address that cannot be used to communicate with the other management server, application failover configuration might fail, or the behavior of a configured application failover environment might become corrupted.

5. Save the nms-cluster-local.properties file.

The com.hp.ov.nms.cluster.interface parameter permits an NNMi administrator to select the communication interface to be used for nnmcluster communication. Make sure that the IP address obtained from the NIC specified in com.hp.ov.nms.cluster.interface matches the IP address that can be resolved from the host name specified in com.hp.ov.nms.cluster.member.hostnames. In an environment where multiple IP addresses are resolved to the same host name, specify the IP address used for application failover communication, not the host name, in the com.hp.ov.nms.cluster.member.hostnames parameter. Specify the com.hp.ov.nms.cluster.member.hostnames parameter.

- Windows: %NnmDataDir%shared\nnm\conf\props\nms-cluster.properties
- Linux: \$NnmDataDir/shared/nnm/conf/props/nms-cluster.properties

18.4 Using the application failover feature

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

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

For details, see the nnmcluster Reference Page.

18.4.1 Application failover behavior

The figure below shows the application failover configuration for two NNMi management servers using the NNMi database. Refer to this figure while reading the rest of this chapter.

Figure 18-1: Application failover configuration (NNMi database)



A database error might result if you remove a standby server from a cluster and run that server as a stand-alone server, and then you add it back into the cluster. If this occurs, run the following command from the command line:

nnmcluster dbsync

^{18.} Configuring NNMi for Application Failover

NNMi 11-00 includes a streaming replication feature within application failover whereby database transactions are sent from the active server to the standby server, keeping the standby server in sync with the active server. This eliminates the need for database transaction logs to be imported to the standby server during failover (as was the case in earlier NNMi versions), thus greatly reducing the time needed for the standby server to take over as the active server. Another benefit of this feature is that database backup files are sent from one node to another only if and when needed. This means that, given the regular transmission of database transaction files, the need for sending large database backup files will normally be infrequent.

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

While both the active and standby servers are running, the active server periodically sends database transaction logs to the standby server. You can modify the frequency of this data transfer by changing the value of the com.hp.ov.nms.cluster.timeout.archive parameter in the nms-cluster.properties file. These transaction logs accumulate on the standby server and are available on the standby server any time it needs to become active.

The standard data transfer frequency is as follows:

- A full backup of the database is transferred every 6 hours.
- Transaction logs (database update information) are transferred every 15 minutes. When a large volume of data in the database is updated, transaction logs are transferred more frequently in some cases. Updates made while data is being transferred are not inherited.

When the standby server receives a full database backup from the active server, it places the information into the NNMi database. The standby server also creates a recovery.conf file to inform the NNMi database that it must incorporate all received transaction logs before it becomes available to other services. If the active server becomes unavailable for any reason, the standby server becomes active by executing the ovstart command to start the NNMi services. The standby NNMi management server imports the transaction logs before starting the remaining NNMi services.

Database files are stored under the following directory:

Windows: %NnmDataDir%shared\nnm\databases\

Linux: \$NnmDataDir/shared/nnm/databases/

In the application failover configuration, three directories (Postgres, Postgres_standby, and Postgres OLD) are created under this directory. These directories are used for the following purposes:

- Postgres: Stores database data received during operation or for a standby purpose.
- Postgres standby: Stores data sent from the active server to the standby server.
- Postgres.OLD: Used by the standby server for saving old Postgres data when new data is received.

If the active server fails, the standby server begins discovery and polling activities. This transition keeps NNMi monitoring and polling your network while you diagnose and repair the failed system.

Important

• NNMi performs resynchronization following a failover in an application failover configuration. This might delay the updating of statuses and incidents.

^{18.} Configuring NNMi for Application Failover

• During resynchronization, a message similar to the following might be displayed, but this is not a problem:

Updating of statuses and incidents was delayed because the Causal Engine queue is full. This might be due to resynchronization performed after a failover, restoration of a backup, or manual resynchronization in an application failover configuration.

18.4.2 Application failover scenarios

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

This subsection explains failover of the NNMi management server because an error occurred under various scenarios.

Table 18-1: Correspondence between possible errors and scenarios

Possible error		Error location	
		Active server	Standby server
Server	Server failure ^{#1}	Scenario 1	Scenario 6
	OS stoppage	Scenario 2	Scenario 6
Process	nnmcluster stoppage	Scenario 3	Scenario 6
	Stoppage of process other than nnmcluster	Scenario 5	N/A ^{#2}
Network	Communication error	Scenario 4	Scenario 4

#1: Server failure assumes a server stoppage caused by a hardware or OS error.

#2: Not applicable because the standby server's NNMi is stopped.

(1) When a failover occurs

In the following scenarios 1 through 3, NNMi fails over to the standby server if automatic failover is enabled and continues to monitor the network:

• Scenario 1: The active NNMi management server fails.

A hardware or OS error causes the active server to stop without going through the OS's shutdown process. The standby server detects that the other server has stopped, goes active to automatically start NNMi, and continues to monitor the network. The original active server runs as a standby server if started.

• Scenario 2: The system administrator shuts down or restarts the active NNMi management server.

The active server has stopped after going through the OS's shutdown process. The standby server detects that the other server has stopped, goes active to automatically start NNMi, and continues to monitor the network. The original active server runs as a standby server if started.

Note that if the NNMi management server is running a Linux operating system and a termination script is run when the OS is shut down, the ovstop command will be executed automatically. This will result in the application failover being disabled and failover will not occur.

• Scenario 3: The NNMi administrator shuts down the cluster.

The administrator or some other cause has stopped the cluster manager (nnmcluster process). The standby server detects that the other server has stopped, goes active to automatically start NNMi, and continues to monitor the network.

Important

If some factor causes only the nnmcluster process of the active server to stop and other NNMi processes continue to remain active, the state is the same as in scenario 3. As a result, NNMi might become active on both the original active server and the new active server. In such a case, recover from the problem by restarting the OS of the original active server.

(2) When no failover occurs

If an event occurs that is not covered by any of the scenarios described in (1) When a failover occurs, failover does not occur. The following are examples:

- Scenario 4: The network connection between the active and the standby NNMi management servers fails. The two servers can no longer communicate with each other. Because heartbeat communications cannot be performed between the cluster managers (nnmcluster process), the following states result:
 - The active server detects that the other server has stopped; the active server continues to run.
 - The standby server also detects that the other server has stopped, goes active, and starts NNMi.

In scenario 4, both NNMi management servers run in the active state. When the network device comes back online, the two NNMi management servers automatically negotiate which server will become the new active server. The other server becomes the standby server and stops NNMi.

The state in which both servers become active causes a problem called *split brain* in HA configurations based on cluster software. However, because application failover uses a different framework, it recovers without any problem, as described below, when communication is restored:

- When communication is restored, one of the servers becomes the standby server, restoring the normal configuration.
- In application failover, the database does not use a shared disk. Instead, the standby server requests the active server to transfer the database and synchronizes it. Therefore, no consistency problem occurs even if NNMi runs on both servers.
- Scenario 5: The NNMi processes have stopped.

Stoppage for any reason of an NNMi process other than the cluster manager (nnmcluster process) does not result in failover.

This is because NNMi processes on the local server are not monitored, although server operations are monitored mutually by the cluster manager's heartbeat communications.

If you want failover to occur whenever an NNMi process stops, employ an HA configuration using cluster software.

• Scenario 6: A failure occurred on the standby server.

An error described in scenarios 1 through 3 (server failure, OS stoppage, or stoppage of the nnmcluster process) has occurred on the standby server. In such a case, the standby server is removed as a cluster configuration member, but NNMi on the active server continues to run and continues to monitor the network.

Important

No report is sent even when the standby server becomes available (resulting in single server operation).

18.4.3 ovstart and ovstop commands used on NNMi management servers configured for application failover

When you use the ovstop and ovstart commands on NNMi management servers configured for application failover, NNMi actually runs the commands listed below. These commands terminate immediately without waiting for completion of the NNMi startup or termination.

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

Note

- When you run the ovstop command, NNMi does not fail over to the standby server. The ovstop command is designed to support temporary stoppage for maintenance. To manually initiate a failover, use the -failover option with the ovstop command. For details, see the *ovstop Reference Page*.
- Note that when you run the ovstop command, automatic failover becomes disabled because the -disable option of nnmcluster is specified. To check whether failover is enabled, run nnmcluster -display to check the Automatic failover column. To enable failover, run nnmcluster -enable.

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

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

Important

If the NNMi management server is running a Linux operating system and a termination script is run when the OS is shut down, the ovstop command will be executed automatically. This will result in the application failover being disabled. To control application failover during maintenance periods, use the nnmcluster -acquire and nnmcluster -relinquish commands before executing the OS shutdown command to set the active and standby servers the way you want them. For details, see the *nnmcluster Reference Page*.

18.4.4 Application failover incidents

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

• NnmClusterStartup: The NNMi cluster was started when there was no active server, so the server was started in the active state. This incident has a NORMAL severity.

^{18.} Configuring NNMi for Application Failover

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

^{18.} Configuring NNMi for Application Failover

18.5 Returning to the original configuration following a failover

If the active node has failed and the standby node is functioning as the active node, you can restore the original configuration after the former active node has been fixed.

To restore the original configuration following a failover:

- 1. Fix the problem at the former active node.
- 2. Run the following command on the desired active node to restore the original configuration:

nnmcluster -acquire

For details, see the nnmcluster Reference Page.

^{18.} Configuring NNMi for Application Failover

18.6 Disabling application failover

Suppose you configure application failover, use it for a few days, and then decide to completely disable it. The following explains how to completely disable application failover. Complete these instructions, which include actions on both the active and standby NNMi management servers configured in the application failover cluster.

- 1. Run the nnmcluster -enable command on the active NNMi management server.
- 2. Run the nnmcluster -shutdown command on the active NNMi management server.
- 3. Wait a few minutes for the old standby NNMi management server to become the new active NNMi management server.
- 4. Run the nnmcluster -display command on the new active (old standby) NNMi management server.
- 5. Search the displayed results for the ACTIVE_NNM_RUNNING status. Repeat step 4 until you see the ACTIVE NNM RUNNING status.
- 6. Run the nnmcluster -shutdown command on the new active (old standby) NNMi management server.
- 7. Run the nnmcluster -display command repeatedly on the new active (old standby) NNMi management server until you no longer see a DAEMON process in any node's type column.
- 8. Edit the following file both NNMi management servers configured in the cluster:

Windows:

%NnmDataDir%shared\nnm\conf\props\nms-cluster.properties

Linux:

\$NnmDataDir/shared/nnm/conf/props/nms-cluster.properties

- 9. Comment out the com.hp.ov.nms.cluster.name option on both NNMi management servers (by adding the characters #! at the beginning of the line) and save each file.
- 10. Edit the following file on both NNMi management servers:

Windows:

```
%NnmDataDir%shared\nnm\databases\Postgres\postgresql.conf
```

Linux:

\$NnmDataDir/shared/nnm/databases/Postgres/postgresql.conf

To edit postgresql.conf, use an editor that can edit files in which only LF $(0 \times 0 A)$ is used as the linefeed code (in Windows, do not use Notepad, use WordPad instead; in Linux, use *vi*).

11. Remove in each file the lines shown below.

Shown here is how these lines might appear on a Windows NNMi management server; they might be slightly different on your server:

```
# The following lines were added by the NNM cluster.
archive_command = 'nnmcluster.exe -archive -logCONFIG "%p" "file:/C:/ProgramData/
Hitachi/Cm2NNMi/shared/nnm/databases/Postgres_standby/TxWALs_send/%f"'
archive_timeout = 900
max_wal_senders = 4
archive_mode = 'on'
wal_level = 'hot_standby'
hot_standby = 'on'
wal_keep_segments = 500
listen_addresses = 'localhost,XX.XX.XX'
```

Make sure to save your changes.

^{18.} Configuring NNMi for Application Failover
- 12. If these are Windows NNMi management servers, navigate to the **Services** (Local) console and do the following on each server:
 - a. Set Startup type for the NNM Cluster Manager to Disabled.
 - b. Set Startup type for the NNM Process Manager to Automatic.
- 13. Create the following trigger file, which tells Postgres to stop running in standby mode and to start running normally:
 - Windows: %NnmDataDir%tmp\postgresTriggerFile
 - Linux: \$NnmDataDir/tmp/postgresTriggerFile
- 14. Run the ovstart command on both NNMi management servers.
- 15. If both NNMi management servers start successfully, remove the following directory from both the standby and active NNMi management servers:
 - Windows: %NnmDataDir%shared\nnm\databases\Postgres_standby
 - Linux: \$NnmDataDir/shared/nnm/databases/Postgres_standby

Note

This directory is a default directory and is the value of the com.hp.ov.nms.cluster.archivedir parameter located in the nmscluster.properties file. These instructions assume you did not change this value. If you changed the value of the com.hp.ov.nms.cluster.archivedir parameter in the nmscluster.properties file, and then remove the directory that equates to the new value.

16. Remove the following directory from both the standby and active NNMi management servers:

- Windows: %NnmDataDir%shared\nnm\databases\Postgres.OLD
- Linux: \$NnmDataDir/shared/nnm/databases/Postgres.OLD

^{18.} Configuring NNMi for Application Failover

18.7 Administrative tasks and application failover

This section explains how to effectively manage application failover when performing administrative tasks such as patching and restarting NNMi management servers.

18.7.1 Upgrading NNMi (including applying a patch)

To upgrade NNMi management servers configured for application failover, see 24.4 Upgrading to NNMi 12-50 configured for application failover.

18.7.2 Starting, stopping, and restarting NNMi

(1) Starting and stopping NNMi

When you use the ovstop or ovstart command to stop or start NNMi configured for application failover, NNMi actually runs nnmcluster commands. For details about the actual commands that run, see 18.4.3 ovstart and ovstop commands used on NNMi management servers configured for application failover.

With NNMi configured for application failover, whether a server becomes the active or standby server is adjusted automatically during the startup. The server on which the nnmcluster command runs first becomes the active server. For details about the behavior during the startup, see 18.4 Using the application failover feature.

When the startup process is completed, the servers go into the states described below for normal operation. To check the server state, run the nnmcluster command without any option (in the interactive mode) or run the nnmcluster -display command, and check the display in the **State** column.

States during normal operation

Active server: ACTIVE_NNM_RUNNING Standby server: STANDBY_READY

When you start NNMi, confirm that the servers go into these states for normal operation. The following table describes the major server states:

State displayed	Role	Explanation
ACTIVE_NNM_STARTING	Active	NNMi is starting.
ACTIVE_DB_BACKUP	Active	NNMi database is being backed up.
ACTIVE_NNM_RUNNING	Active	NNMi is running.
STANDBY_RECV_DBZIP	Standby	Database is being transferred from the NNMi of the active server.
STANDBY_READY	Standby	Server with NNMi is ready as the standby server.

Before performing an application failover, for example switching between the active and standby servers, make sure that the servers are in the states for normal operation. If you perform a failover before the servers go into these states, the servers cannot be correctly synchronized, and as a result NNMi might fail to start and will go into the ACTIVE_NNM_FAILED state. In this case, stop both servers and restart NNMi. If a database problem prevents NNMi from starting, reset the database, restore the backup data, and then restart NNMi.

^{18.} Configuring NNMi for Application Failover

(2) Restarting NNMi

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

To restart either the active or the standby NNMi management server:

- 1. Run the nnmcluster -disable command on the NNMi management server to disable the application failover feature.
- 2. Restart the NNMi management server by executing the following commands:

```
ovstop
ovstart
```

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

Important

For important information about NNMi's TrapReceiver process and how it relates to failovers, see 21.16 NNMi NmsTrapReceiver process.

Application failover control after a communication failure

Once a communication failure between the two cluster nodes has been resolved, the NNMi management server that had been running the longest before the communication failure occurred (that is, the previous active NNMi management server) is designated as the active server.

(3) NNMi failover

When a system configured for application failover fails, a failover automatically occurs according to a scenario described in 18.4.2 Application failover scenarios and NNMi starts on the active server.

To manually initiate a failover:

1. Run the ovstop -failover command on the active server.

After NNMi stops, the cluster manager (nnmcluster) stops and the standby server becomes the new active server, and then NNMi starts.

2. When step 1 is performed, the original active server is removed as a cluster configuration member. To bring back this server into the cluster as a standby server, run the ovstart command.

The ovstart command is replaced with the nnmcluster -daemon command before being executed.

18.7.3 Backing up and restoring NNMi

(1) Backing up NNMi

You can back up NNMi configured for application failover by using the same procedure as is used for ordinary systems. However, you cannot use the -force option (for forcing NNMi into a state suitable to backup). Place NNMi in a state appropriate for backing up before starting backup.

Do the following on the active server:

^{18.} Configuring NNMi for Application Failover

1. Place NNMi in a state appropriate for backing up.

Using nnmbackup.ovpl:

For online backup: Start the NNMi service.

For offline backup: Stop the NNMi service.

Using nnmbackupembdb.ovpl:

Start the NNMi service.

2. Make a backup.

Run the nnmbackup.ovpl or nnmbackupembdb.ovpl command.

Important

- Make the backup on the active server (the server that was previously running if you stopped NNMi for an offline backup).
- Make the backup on both the active server and standby server if restoring NNMi failover environment on a different set of servers.

(2) Restoring NNMi

To restore your NNMi database from an original backup when active and standby NNMi management servers are configured for application failover, follow these steps:

1. Run the nnmcluster -halt command on the active NNMi management server.

This stops NNMi on both the active and standby servers. To verify that the NNMi service has stopped, run the nnmcluster command with no options specified (interactive mode) or the nnmcluster -display command.

2. Delete or move the following directories on both the active and standby NNMi management servers:

• Windows:

%NnmDataDir%shared\nnm\databases\Postgres_standby %NnmDataDir%shared\nnm\databases\Postgres.OLD

• Linux:

\$NnmDataDir/shared/nnm/databases/Postgres_standby
\$NnmDataDir/shared/nnm/databases/Postgres.OLD

- 3. Restore the database on the active NNMi management server:
 - a. Temporarily cancel the application failover configuration setting.

Modify the following file to comment out the cluster name com.hp.ov.nms.cluster.name (by adding the characters #! at the beginning of the line):

Windows:

%NnmDataDir%shared\nnm\conf\props\nms-cluster.properties

Linux:

\$NnmDataDir/shared/nnm/conf/props/nms-cluster.properties

b. Restore the database as normal. Run the nnmrestore.ovpl command or the nnmrestoreembdb.ovpl command with the -force option. After the services necessary for restoration are started by the specification of the -force option, restoration is performed. For details about these commands, see 20.3 Restoring NNMi data.

If you restore data that was backed up using nnmbackup.ovpl, the change made in step a might be overwritten in the restored files. Therefore, do step a again.

- c. Run the ovstop command on the active NNMi management server. This stops the services that were started in step b for restoration processing.
- d. Reset the application failover configuration.

Modify the following file to uncomment the cluster name com.hp.ov.nms.cluster.name (by deleting the #! characters added in step a):

Windows:

%NnmDataDir%shared\nnm\conf\props\nms-cluster.properties

Linux:

\$NnmDataDir/shared/nnm/conf/props/nms-cluster.properties

- 4. Run the ovstart command on the active NNMi management server.
- 5. Wait until the active NNMi management server generates a new backup (a ZIP file for synchronizing the active and standby servers).

To verify that this step is complete, run the nnmcluster -display command and look for an ACTIVE NNM RUNNING message.

6. Run the ovstart command on the standby NNMi management server.

The standby NNMi management server copies and extracts the new backup (the ZIP file created in step 5). To verify that this step is complete, run the nnmcluster -display command and look for a STANDBY_READY message.

(3) Restoring NNMi Failover Environment on a different set of servers

Restoring NNMi failover environment on a different set of servers requires obtaining backup of both NNMi active and standby systems, restoring them on the required servers, and also changing the hostnames in certain property files.

To restore NNMi failover environments, follow these steps:

- 1. Obtain a complete offline backup of all NNMi data on both Active and Standby systems in the source failover environment. For more information, see 20.2 Backing up NNMi data.
- 2. Copy the backup files to the respective destination Active and Standby systems.
- 3. Install NNMi to the same version and patch level as were in place for the backup.
- 4. Restore NNMi data on both Active and Standby systems.

Perform the step 3 described in (2) Restoring NNMi on both Active and Standby systems.

- 5. On both active and standby NNMi management servers, do the following:
 - a. Identify hostnames of both active and standby NNMi management servers.
 - b. Open the following file.

Windows:

%NnmDataDir%shared\nnm\conf\props\nms-cluster.properties

Linux:

\$NnmDataDir/shared/nnm/conf/props/nms-cluster.properties

c. Add the hostnames of both active and standby nodes to the

com.hp.ov.nms.cluster.member.hostnames parameter.

com.hp.ov.nms.cluster.member.hostnames = fqdn_for_active, fqdn_for_standby

6. Configure NNMi failover environment to use SSL certificates for secure communication. For more information, see 10. Working with Certificates for NNMi.

18.7.4 Modifying the NNMi settings

(1) Modifying the settings

This subsection explains how to modify the NNMi settings in the application failover configuration.

(a) NNMi settings files

When modifying the NNMi settings files, make the same modification on both the active and standby servers so that the files are identical in content.

Important

If a setting modification will result in a restart of NNMi, stop both servers before modifying the setting.

To modify settings without stopping NNMi operation, follow the procedure described below. In each step, run the nnmcluster -display command to verify that the processing has been completed before proceeding to the next step.

1. Run the ovstop -failover command on server A.

Server A stops and server B becomes active.

- 2. Modify the settings on server A.
- 3. Run the ovstart command on server A to start it as the standby server.
- 4. Run the ovstop -failover command on server B.

Server B stops and server A becomes active.

- 5. Modify the settings on server B.
- 6. Run the ovstart command on server B to start it as the standby server.

(b) NNMi database

Because the active and standby servers automatically maintain synchronization with each other on the NNMi database, no operation by the system administrator is required.

For details about this behavior, see 18.4 Using the application failover feature.

(2) Database reset

To perform the database reset explained in 4.8 Resetting the NNMi configuration and database, you must cancel the application failover configuration temporarily by doing the following:

1. (Optional) If you wish to save the current NNMi settings, do the following on the active server:

- Use the nnmconfigexport.ovpl command to output the NNMi settings to an XML file.
- Use the nnmtrimincidents.ovpl command to archive the NNMi incidents.
- 2. Run the nnmcluster -halt command on the active server.

NNMi stops on both the active and standby servers.

To verify that the NNMi service has stopped, run the nnmcluster command with no options specified (interactive mode) or the nnmcluster -display command.

3. Reset the NNMi database on the active server.

a. Temporarily cancel the application failover configuration settings.

Modify the following file to comment out the cluster name com.hp.ov.nms.cluster.name (by adding the characters #! at the beginning of the line):

Windows:

```
%NnmDataDir%shared\nnm\conf\props\nms-cluster.properties
```

Linux:

\$NnmDataDir/shared/nnm/conf/props/nms-cluster.properties

b. (Optional) Before the database data is deleted, use the following command to back up the existing database as needed:

nnmbackup.ovpl -type offline -target backup-directory

c. Delete the NNMi database and then re-create it:

nnmresetembdb.ovpl-nostart

d. Run the ovstop command on the active NNMi management server. The service that was started in step c stops.

e. Reconfigure application failover.

Modify the following file to uncomment the cluster name com.hp.ov.nms.cluster.name (by deleting the #! characters added in step a):

Windows:

```
%NnmDataDir%shared\nnm\conf\props\nms-cluster.properties
```

Linux:

\$NnmDataDir/shared/nnm/conf/props/nms-cluster.properties

4. Delete or move the following directories on both the active and standby servers:

• Windows:

%NnmDataDir%shared\nnm\databases\Postgres_standby %NnmDataDir%shared\nnm\databases\Postgres.OLD

• Linux:

\$NnmDataDir/shared/nnm/databases/Postgres_standby
\$NnmDataDir/shared/nnm/databases/Postgres.OLD

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

To verify that this step is complete, run the nnmcluster -display command and look for an ACTIVE_NNM_RUNNING message.

6. Run the ovstart command on the standby NNMi management server.

To verify that this step is complete, run the nnmcluster -display command and look for a STANDBY_READY message.

This leaves the NNMi database in the default settings only.

Set up NNMi on the active server. To import the NNMi settings saved in step 1, use the nnmconfigimport.ovpl command.

^{18.} Configuring NNMi for Application Failover

18.7.5 Changing the NNMi database password

- 1. Perform the procedure described in 18.6 Disabling application failover to temporarily disable the application failover configuration.
- 2. On each NNMi management server, change the password. For details about the procedure, see the *nnmchangeembdbpw.ovpl Reference Page*.
- 3. Perform the procedure described in 18.3.2 Configuring application failover with the NNMi Cluster Setup Wizard to reactivate the application failover configuration.

^{18.} Configuring NNMi for Application Failover

18.8 Network latency/bandwidth considerations

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

The NNMi database can become quite large and can grow to 1GB or more even though this file is always compressed. Also, NNMi generates hundreds, or even thousands, of transaction logs during the built-in backup interval (a configuration parameter that defaults to six hours). Each transaction log can be several megabytes, up to a maximum size of 16 MB (these files are also compressed). Example data collected from a Hitachi test environments is shown here:

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

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

18.8.1 Application failover and the NNMi database

After you configure NNMi using the database for application failover, NNMi does the following:

- 1. The active server performs a database backup, storing the data in a single ZIP file.
- 2. NNMi sends this ZIP file across the network to the standby server.
- 3. The standby server extracts the ZIP file and configures the database to import transaction logs at the first startup.
- 4. The database on the active server generates transaction logs, depending on database activity.
- 5. Application failover sends the transaction logs across the network to the standby server, where they accumulate on the disk.
- 6. When the standby server becomes active, NNMi starts and the database imports all transaction logs across the network.

The amount of time this takes depends on the number of files and the complexity of the information stored within those files.

- 7. After the standby server imports all the transaction logs, the database becomes available and the standby server starts the remaining NNMi processes.
- 8. The original standby server is now active, and the procedure starts over at step 1.

(1) Network traffic in an application failover environment

NNMi transfers many items across the network from the active server to the standby server in an application failover environment:

- Database activity (the database backup as a single ZIP file)
- Transaction logs
- A periodic heartbeat so that each application failover node verifies that the other node is still running.
- File comparison lists so that the standby server can verify that its files are in synchronization with those on the active server
- Miscellaneous events, such as changes in parameters (enable/disable failover and others) and nodes joining or leaving the cluster

The first two items generate 99% of the network traffic used by application failover. This subsection explores these two items in more detail.

Database activity

NNMi generates transaction logs for all database activity.

Database activity includes everything in NNMi. This activity includes, but is not limited to, the following database activities:

- Discovering new nodes
- Discovering attributes about nodes, interfaces, VLANs, and other managed objects
- State polling and status changes
- Incidents, events, and root cause analysis
- Operator actions on the NNMi console

Database activity is outside of your control. For example, an outage on the network results in NNMi generating many incidents and events. These incidents and events trigger state polling of devices on the network, resulting in updates to device status in NNMi. When the outage is restored, additional node up incidents result in further status changes. All of this activity updates entries in the NNMi database.

Although the NNMi database itself grows with database activity, it will reach a stable size for your environment and will exhibit only moderate growth over time.

Database transaction logs

The NNMi database works by creating an empty 16-megabyte file, and then writing database transaction information into that file. NNMi closes this file, and then makes it available to application failover after 15 minutes or after writing 16 megabytes of data to the file, whichever comes first. That means that a completely idle database will generate one transaction log file every 15 minutes, and this file will be essentially empty. Application failover compresses all transaction logs, so an empty 16-megabyte file compresses down to under 1 megabyte. A full 16-megabyte file compresses to about 8 megabytes. Keep in mind that during periods of higher database activity, application failover generates more transaction logs in a shorter period of time, because each file gets full faster.

(2) An application failover traffic test

The following test resulted in an average of about two transaction log files per minute, with an average file size of 7 megabytes per file. This was due to the database activity associated with discovery of an additional 5,000 nodes added with each failover event. The database in this test case eventually stabilized at about 1.1 gigabytes (as measured by the size of the backup ZIP file), with 31,000 nodes and 960,000 interfaces.

^{18.} Configuring NNMi for Application Failover

Testing method

During the first 4 hours, test personnel seeded NNMi with 5,000 nodes and waited until discovery stabilized. After 4 hours, test personnel induced failover (the standby server became active, and the previously active server became the standby). Immediately after failover, test personnel added approximately 5,000 more nodes, waited another 4 hours to let the NNMi discovery process stabilize, and then induced another failover (failed back to the original active server).

Test personnel repeated this cycle several times with some variation in the time between failovers (4 hours, then 6 hours, then 2 hours). After each failover event, test personnel measured the following:

- Size of the database created when the node first became active
- Size of the backup ZIP file
- Transaction logs
- Total number of files and the amount of disk space used
- Number of nodes and interfaces in the NNMi database immediately before inducing failover
- Elapsed time to complete failover

This was the time from the initial ovstop command on the active server until the standby server became fully active with NNMi running.

Results

The following table summarizes the results:

Table 18-2:	Application	failover	test	results

Hours	DB.zip size (MB)	No. of transaction logs	Transaction log size (GB)	Nodes	Interfaces	Failover time (minutes)
4	6.5	50	0.3	5,000	15,000	5
8	34	500	2.5	12,000	222,000	10
12	243	500	2.5	17,000	370,000	25
16	400	500	3.5	21,500	477,000	23
20	498	500	3.5	25,500	588,000	32
26	618	1,100	7.5	30,600	776,000	30
28	840	400	2.2	30,600	791,000	31
30	887	500	2.5	30,700	800,000	16

Observations

When NNMi transferred files from the active server to the standby server, the transfer averaged about 5 gigabytes every 4 hours, which is a continuous throughput of approximately 350 kilobytes/sec or 2.8 megabits/sec.

Note

- This data does not include any other application failover traffic, such as the heartbeat, file consistency checks, or other application failover communications. This data also excludes the overhead of network I/O, such as packet headers. This data only included the actual network payload of each file's contents moving across the network.
- The traffic volume generated by an NNMi application failover environment is extremely large. Application failover identifies new transaction logs on the active server every five minutes and sends these logs to the standby server. Depending on network speed, the standby server might receive all

^{18.} Configuring NNMi for Application Failover

the new files in a short time, resulting in a relatively idle network for the remainder of that 5-minute interval.

Each time the active and standby servers switch roles (the standby server becomes active and the active server becomes standby), the new active server generates a complete database backup that it sends across the network to the new standby server. Such a database backup also occurs periodically, backing up every 24 hours by default. Each time NNMi generates a new backup, it sends the backup to the standby server. Having this new backup available on the standby server reduces the failover time, as all transaction logs NNMi generated in that 24-hour interval are already in the database and do not need to be imported at the time of a failover.

The information provided in this section will help you understand how the network might perform after a failover when using NNMi with application failover using the NNMi database.

^{18.} Configuring NNMi for Application Failover

19

Configuring NNMi in a High Availability Cluster

High availability (HA) refers to a hardware and software configuration that provides for uninterrupted service in the event some aspect of the running configuration fails. An HA cluster defines a grouping of hardware and software that works together to ensure continuity in functionality and data when failover occurs. This chapter provides a template for configuring NNMi to run in an HA environment. This chapter 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 commands specific to the HA product where these instructions specify NNMi-provided commands.

19.1 HA concepts

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



Figure 19-1: Architecture of a high availability cluster

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

In modern cluster environments, such as Veritas Cluster Server, Symantec Cluster Server, or Windows Server Failover Cluster, 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. The following figure shows an example HA resource group.

Figure 19-2: Typical HA resource group layout



This manual 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. The following table lists for each supported HA product the term that equates to *HA resource group* in this manual.

Table 19-1: Terminology for HA resource	group in the supported HA products
--	------------------------------------

HA product	Abbreviation	Equivalent term for HA resource group
Veritas Cluster Server	VCS	Service group
Symantec Cluster Server	SCS	Service group
Windows Server Failover Cluster	WSFC [#]	Resource group

^{19.} Configuring NNMi in a High Availability Cluster

HA product	Abbreviation	Equivalent term for HA resource group
HA Monitor	HA Monitor	Server

#

WSFC is also referred to as MSFC (Microsoft Failover Cluster), but in this manual WSFC is used.

Not all the HA products listed in Table 19-1 are supported by all OSs.

For the supported cluster software and versions, visit Hitachi's home page.

19.1.1 HA terms

The following table lists and defines some common HA terms.

Table 19-2:	Common	HA	terms
-------------	--------	----	-------

Term	Description
HA resource group	A group of resources running in a cluster environment (under an HA product).
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, <i>and</i> the first system on which HA is configured. The shared disk is mounted on the primary cluster node for initial setup. 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 the passive cluster nodes, which then becomes the active cluster node for that HA resource group.

19.1.2 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 installed 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.)

Note

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

NNMi-only scenario

The figure below shows a graphical representation of the NNMi HA cluster scenario.

^{19.} Configuring NNMi in a High Availability Cluster

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

Figure 19-3: Basic scenario for NNMi HA cluster



For details about how to implement this scenario, see 19.4.2 Configuring NNMi for HA.

19.1.3 Manpages

NNMi provides the following manpages to assist you with an NNMi High Availability configuration:

- 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.

^{19.} Configuring NNMi in a High Availability Cluster

19.2 Verifying the prerequisites to configuring NNMi for HA

An HA cluster in which NNMi is to be run must satisfy the requirements shown in this section.

Overall system configuration

- NNMi cannot be used in a configuration that includes multiple HA products. NNMi might not be able to correctly recognize the applicable HA product, resulting in a malfunction.
 - Windows: The NNMi installation target (%NnmDataDir% and %NnmInstallDir%) must be the same in all cluster nodes.
- Both NNMi management server must be running the NNMi of the same version (including a version of the modified version).
- In Windows: NNMi uses the cluster.exe command in the cluster system. However, if you use Windows Server 2012 or later, this command is not installed by default.

To install the command, from Server Manager > Add Roles and Features, select Features > Remote Server Administration Tools > Feature Administration Tools > Failover Clustering Tools > Failover Cluster Command Interface.

- Refer to the following items in the *Release Notes* described in the table of *Chapter 4. Memory and Disk Space Requirements* for disk space requirements.
 - Disk space of the local disk

The value of "Disk space for application installation" and the value of "Disk space for database and data during execution"

• Disk space of the shared disk

The value of "Disk space for database and data during execution"

Resource group

- NNMi setup must be initiated when there is no resource group to be configured. NNMi cannot be added to an existing resource group.
- Make sure that the configuration supports virtual IP addresses and shared disks and that it allows NNMi to access them.

Shared disk

- NNMi's shared disk is stored at the location shown below. The directory name cannot contain any spaces; it must always be NNM.
 - Windows: *drive-letter*: \NNM (example: Y: \NNM) or *drive-letter*: *any-directory*\NNM (example: Y: \JP1\NNM)
 - Linux: *mount-point*/NNM (example: /shdsk1/NNM)
- For a shared disk, use a storage connected by Fibre (FC-SAN), SCSI, or iSCSI. NNMi does not support a configuration that uses NAS with NFS or CIFS connection.
- In Windows: For a shared disk, use a disk with a drive letter assigned. Do not use a disk that has been mounted by using the mount settings in **Disk Manager** or the mountvol command. In this manual, information regarding *mount* applies to Linux.
- In Windows: In Windows Server, dynamic disks are not supported for clustering using Microsoft Cluster Service.

Virtual IP addresses

• Configure virtual IP addresses and virtual host names in such a manner that host names can be converted to IP addresses and vice versa for a name service such as DNS and hosts file.

^{19.} Configuring NNMi in a High Availability Cluster

- Even if you use a name service such as DNS, the virtual IP addresses and virtual host names must be configured in the hosts file in such a manner that the names can be resolved. This prevents a failover error resulting from a name resolution error in the event of a communication error that leads to failover.
- If you configure IPv6's logical IP address as a resource, add it manually after you have finished the procedure described in 19.4 Configuring HA. For the configuration procedures, see the cluster software documentation. The cluster software versions that support IPv6, a configuration using IPv6, and whether IPv6 and IPv4 can be intermixed depend on the cluster software specifications.

Virtual host names

When you configure your cluster environment, set virtual host names to be resolved as IPv4 addresses.

^{19.} Configuring NNMi in a High Availability Cluster

This section provides notes about HA configurations.

19.3.1 Notes about using related products

Note the following when you use an NNMi-related product (such as JP1/SNMP System Observer (JP1/SSO)):

- Set up NNMi first, and then set up the related product.
- Register NNMi and the JP1/SSO related product into the same resource group. The resource dependency that is set in the cluster software must be as follows:
 - For JP1/SSO, set the dependency so that NNMi is required.
 - For NNMi, set the dependency so that the shared disk and virtual IP address are required. In Windows, also set the dependency for network name resources.

For details about how to configure related products, see the applicable documentation, *Release Notes*, and instruction manuals.

19.3.2 Notes about configuration tasks and operations

Note the following when you configure and run NNMi for HA:

- The OS user who runs NNMi must have the permissions needed to perform all cluster software operations. The OS user must be able to perform various operations using the cluster software, including creation of NNMi resources and starting and stopping resource groups.
- Operations must be performed with the cluster software running. Various NNMi commands for HA configuration are used on the cluster software to perform processes, such as configuration and verification of configuration. If the cluster software is not running, these processes will result in an error.
- If you restart the NNMi service using the procedure described in this manual and in the *Release Notes*, set NNMi in the maintenance mode in the cluster environment before restarting the NNMi service, unless instructed otherwise.
- When you execute commands and edit local files, make sure that the NNMi resource group is online, unless instructed otherwise in the documentation.

Also make sure that failover will not occur within three minutes of a command's execution or of local file editing. If a command is executed or a local file is edited while the resource group is offline and failover occurs within three minutes of that operation, the settings might be overwritten by the old configuration.

• The NNMi resources assume that failover will occur in the event of a failure. Configure NNMi to result in a failover if a failure occurs in the resource group; do not restart NNMi on the system resulting in a failure.

For details about the configuration method, see *Help* for the cluster software.

- In an environment that a failover is occurred by a failure, when you perform a failback, perform a failback after you recover all failures and stop all NNMi processes.
- In an HA environment, NNMi database is corrupted only occasionally due to process down and disk failure. Therefore back up NNMi settings and data periodically.

^{19.} Configuring NNMi in a High Availability Cluster

19.3.3 Other notes

- Depending on the environment, it might take 10 minutes or more to start the NNMi services.
- Windows: Neither Online Pending nor Offline Pending is displayed as the status of *resource-group*-APP on the Failover Cluster Management console. To make sure that *resource-group*-APP is in wait status, verify that the following status is Pending on the Failover Cluster Management console:
 - *cluster-name* > **Services and Applications** > *resource_group* > Summary of *resource_group* > Status
- Windows: When you use the data collection tool, create cluster.log by executing cluster.exe log /g.

^{19.} Configuring NNMi in a High Availability Cluster

19.4 Configuring HA

This section describes the procedures for configuring a new HA configuration for NNMi.

19.4.1 Configuring NNMi certificates for HA

The NNMi installation process configures a self-signed certificate for secure communications between the NNMi console and the NNMi database. The process for configuring NNMi for HA correctly shares the self-signed certificate among the primary and secondary cluster nodes. You do not need to take any extra steps to use the default certificate with NNMi running under HA.

If you want to use a different self-signed certificate or a Certificate Authority (CA)-signed certificate for NNMi communications, you must do some additional work. After obtaining the new certificate, complete the steps shown in 10.3.6 Working with Certificates in High-Availability Environments. You can complete this procedure before or after configuring NNMi for HA.

19.4.2 Configuring NNMi for HA

This subsection explains the procedure for configuring NNMi for HA and the settings that must be determined at the evaluation stage.

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

1. Copy the NNMi data files to the shared disk.

Perform this task on the primary cluster node.

- 2. Configure NNMi to run under HA.
 - Perform this task on the primary cluster node.
 - Perform this task on the secondary cluster node.

The following shows the configuration procedure.

19. Configuring NNMi in a High Availability Cluster



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



Important

If you use HTTPS communication to access NNMi servers, you must set up to use a certificate before you verify startup of the primary cluster node. For details, see 10.3.6 Working with Certificates in High-Availability Environments.

Note

- You cannot configure NNMi for HA simultaneously on multiple cluster nodes. In a cluster environment, configure nodes for HA one by one, so that the HA setting process on the primary cluster node finishes before the HA setting process on the secondary cluster node starts.
- For HA Monitor, you perform the configuration procedure without using nnmhaconfigure.ovpl. For details about the configuration procedure, see the *Release Notes*.

During failover, the NNMi console is unresponsive. After failover completes, NNMi users must sign in to continue their NNMi console sessions.

For important information about NNMi's TrapReceiver process and how it relates to failovers, 21.16 NNMi NmsTrapReceiver process.

^{19.} Configuring NNMi in a High Availability Cluster

(1) NNMi HA configuration information

The HA configuration script (nnmhaconfigure.ovpl) collects information about the NNMi HA resource group. The table below lists and describes the information that you will need for configuring the primary cluster node. Gather this information before you begin the configuration procedure.

You will enter this information interactively during the configuration process by running an HA configuration script (nnmhaconfigure.ovpl). The information requested depends on the OS in use, the type of HA product, and the system configuration. Enter the requested information by following the instructions.

HA configuration item	Description
HA resource group	The name of the resource group for the HA cluster that contains NNMi. This name must be unique, specific to NNMi, and not currently in use. Example: nnmtest1
	Note: The HA resource group name cannot contain any space characters.
	Note: The character set and the number of characters permitted for the name depend on the cluster software specifications. For details, see the information provided following this table.
	Note: Make sure that the HA resource group name is unique among all other resource names and resource group names (including any substring of a resource name or a resource group name). For example, if there is a resource group named testA, a name containing any substring of testA, such as test or est, cannot be used.
	Note: Once the HA resource group has been created, it cannot be renamed. To rename the HA resource group, you must cancel HA configuration for NNMi, and then perform HA configuration again using a new name.
Virtual host name	The name of the virtual host. This is the short name, not the FQDN name containing the domain name. This host name must map to the virtual IP address for the HA resource group. The virtual host short name and the virtual IP address must be resolvable.
	Note: The virtual host cannot be renamed after the HA configuration has been completed. If you want to rename the virtual host, cancel HA configuration for NNMi, and then perform HA configuration again a new name.
	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, we recommend 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 Linux 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, which must be an IPv4 address.
Virtual host network interface	The network interface on which the virtual host IP address is running. Examples:
	Windows: Local Area ConnectionLinux: eth0
Shared file system type	The type of shared disk configuration being used for the HA resource group. The following are the possible values:
	• 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 details, see the <i>File system type</i> entry in this table.
	• none: The shared disk uses a configuration other than that described for the disk option, such as SAN or NFS. The HA configuration script configures the shared disk.
	Note: Because JP1/NNMi does not support the operation when none is specified, you must specify disk.
File system type	(Linux only)

Table 19-3: NNMi HA primary cluster node configuration information

^{19.} Configuring NNMi in a High Availability Cluster

HA configuration item	Description
File system type	The file system type of the shared disk (when 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. The following shared disk formats have been tested: • ext2, ext3, and vxfs for VCS or SCS
Disk group	(Linux only) The name of the disk group for the NNMi shared file system. Example: shdg01
Volume group	(Linux only) The name of the volume group for the NNMi shared file system. Example: vg03
Mount directory (mount point)	The directory location for mounting the NNMi shared disk. This mount point must be consistent among systems (that is, all nodes must use the same name for the mount point). In Windows, specify <i>drive-letter</i> or <i>drive-letter</i> : \ <i>any-directory</i> . The directory name cannot contain any spaces.
	Example:
	• Windows: Y: or Y:\JP1
	• Linux: /nnmmount
	Note: NNMi's shared data is stored in the directory named NNM that is created immediately below the directory specified here (path of the storage directory is shown below). The storage directory name (NNM) is fixed.
	• Windows: <i>drive-letter</i> : \NNM or <i>drive-letter</i> : \ <i>any-directory</i> \NNM
	• Linux: mount-point/NNM
	Note: You will not be able to change the mount point after HA configuration for the relevant NNMi instance is complete. In a Windows environment, you will also not be able to change the drive letter of the mount point after HA configuration for the relevant NNMi instance is complete. If it is necessary to change the mount point, perform the procedure in 19.7 Unconfiguring NNMi from an HA cluster to undo the HA configuration, and then perform the procedure in 19.4 Configuring HA to configure HA again.

The character set and the number of characters permitted for the name of the HA resource group for NNMi must comply with the cluster specifications. The following characters are available to name your HA resource group for NNMi:

- In Windows WSFC
 - Characters:
 - Alphabetic characters (a to z, A to Z), numeric characters (0 to 9), hyphen (-), underscore (_), period (.)
 - Number of characters: Maximum of 247 characters including the path name %NnmDataDir%hacluster \resource-group
- In Linux VCS or Linux SCS
 - Characters:

Alphabetic characters (a to z, A to Z), numeric characters (0 to 9), hyphen (-), underscore (_) The name must begin with an alphabetic character.

- Number of characters: Maximum of 255 characters
- In Linux HA Monitor
 - Characters:

Alphabetic characters (a to z, A to Z), numeric characters (0 to 9)

The name must begin with an alphabetic character.

^{19.} Configuring NNMi in a High Availability Cluster

• Number of characters: Maximum of 8 characters

19.4.3 Configuring NNMi for HA (Windows)

This subsection explains how to configure NNMi for HA in a Windows environment.

In HA configuration for NNMi, you create a new resource group for NNMi. Therefore, you must begin the configuration procedure when there is no resource group to be configured.

The script (nnmhaconfigure.ovpl) used to configure NNMi for HA internally creates a resource group and individual resources for the cluster software. When the configuration procedure is completed, the following resource group has been configured.

Table 19-4: Components of resource group for NNMi in WSFC

Resource name	Resource type	Description
virtual-host-name	Network name	Controls virtual host names.
resource_group-IP	IP address	Controls virtual IP addresses.
<pre>resource_group-mount</pre>	Physical disk	Controls the shared disk.
resource-group-APP	General-purpose script	Controls the start, stop, and monitoring of NNMi.

In WSFC, nnmhaconfigure.ovpl configures the above resources by internally executing commands such as cluster.exe.

- *resource-group* is replaced with the HA resource group name.
- Set the resource dependency so that the general-purpose script resource *resource-group*-APP requires *IP-address-resource*, *disk-resource*, and *network-name-resource*.

(1) Example of settings for each resource in WSFC

This subsection provides an example of the settings for each resource in WSFC when the configuration is completed. Replace *resource-group* with the actual HA resource group name.

Item	Details
General	Resource name: virtual-host-name
	Resource type: Network name
	DNS name: virtual-host-name
	• Full name: virtual-host-name.test.com
	• Network: 192.168.100.0/24
	• IP address: 192.168.100.24
	• NetBIOS status: OK
	DNS status: OK
	• kerberos status: OK
Dependencies	IP-address-resource
Policies	• If all the restart attempts fail, begin restarting again after the specified period (<i>hh:mm</i>) is enabled. Period: 15:00

Table 19-5: network-name-resource

^{19.} Configuring NNMi in a High Availability Cluster

Item	Details
Policies	 Maximum restarts in the specified period: 0 If restart is unsuccessful, fail over all resources in this service or application is enabled. Period: 03:00
Advanced Policies	 Basic resource health check interval: Use standard time period for the resource type Thorough resource health check interval: Use the standard time period for the resource type Run this resource in a separate Resource Monitor is disabled.

Table 19-6:	IP-address-resource

Item	Details	
General	Resource name: <i>resource-group</i> -IP	
	Resource type: IP address	
	• Network: 192.168.100.0/24	
	• Static IP address: 192.168.100.24 [#]	
	• Enable NetBIOS for this address is enabled.	
Dependencies	No dependencies	
Policies	• If all the restart attempts fail, begin restarting again after the specified period (hh:mm) is enabled.	
	Period : 15:00	
	Maximum restarts in the specified period: 0	
	• If restart is unsuccessful, fail over all resources in this service or application is enabled.	
	Period : 03:00	
Advanced Policies	Basic resource health check interval: Use the standard time period for the resource type	
	• Thorough resource health check interval: Use the standard time period for the resource type	
	Possible owners is disabled.	

#: DHCP is not enabled.

Table 19-7: physical-disk-resource

Item	Details
General	 Resource name: resource_group-mount Resource type: Physical disk Volume: Y:
Dependencies	No dependencies
Policies	 If all the restart attempts fail, begin restarting again after the specified period (<i>hh:mm</i>) is enabled. Period: 15:00 Maximum restarts in the specified period: 0 If restart is unsuccessful, fail over all resources in this service or application is enabled. Period: 03:00
Advanced Policies	 Basic resource health check interval: Use the standard time period for the resource type Thorough resource health check interval: Use the standard time period for the resource type Possible owners is disabled.

Table 19-8: general-purpose-script-resource

Item	Details
General	Resource name: <i>resource-group</i> -APP

19. Configuring NNMi in a High Availability Cluster

Item	Details
General	 Resource type: General-purpose script Script path[#]: %NnmDataDir%hacluster/<i>resource-group</i>/hamscs.vbs
Dependencies	network-name-resource, IP-address-resource, and disk-resource
Policies	 If all the restart attempts fail, begin restarting again after the specified period (<i>hh:mm</i>) is enabled. Period: 15:00 Maximum restarts in the specified period: 0 If restart is unsuccessful, fail over all resources in this service or application is enabled. Period: 30:00
Advanced Policies	 Basic resource health check interval: Use the standard time period for the resource type Thorough resource health check interval: Use the standard time period for the resource type Possible owners is enabled.

#

For the script path, the full path with the environment variable expanded is set. Example:

C:/ProgramData/Hitachi/Cm2NNMi/hacluster/jp1ha1/hamscs.vbs

(2) Configuring NNMi on the primary cluster node

Complete the procedure described below on the primary cluster node.

(a) Preparations

To start with the preparations:

- 1. If you have not already done so, complete the procedure described in 19.2 Verifying the prerequisites to configuring NNMi for HA.
- 2. If you have not already done so, install NNMi, and then verify that NNMi is working correctly.
- 3. Use the following command to back up all NNMi settings and data:

Example:

nnmbackup.ovpl-scopeall-targetnnmi_backups

For details about this command, see 20. NNMi Backup and Restore Tools.

In the initial status of NNMi cluster environment configuration, the data in the primary cluster node must exactly match the data in the secondary cluster node. Therefore, restore the backup data obtained here during the secondary cluster node configuration procedure.

(b) Copying data to the shared disk

Next, copy data to the shared disk.

1. Provide a shared disk for the NNMi HA resource group.

Use Windows Explorer and the Disk Management tool to assign a drive letter.

Important

Verify that the provided shared disk satisfies the following conditions:

^{19.} Configuring NNMi in a High Availability Cluster

• Use the Disk Management tool to make sure that the shared disk displays online. If reserved is displayed, it indicates that WSFC has control of the shared disk. Use the **Delete** action from the WSFC user interface to remove the shared disk from WSFC control. Also use the Disk Management tool to confirm that the reserve flag has changed to online.

- It has already been formatted.
- It has enough free space.
- It is not being used by any other resource group.
- A user with administrator permissions has the **Full Control** permission, and a built-in Local Service user (Users group) has the **Read & execute** permissions.

2. Stop NNMi:

```
%NnmInstallDir%bin\ovstop -c
net stop NnmTrapReceiver
```

3. Copy the NNMi files to the shared disk:

```
%NnmInstallDir%misc\nnm\ha\nnmhadisk.ovpl NNM -to HA-mount-point
```

Important

Specify for *HA-mount-point* the drive of the shared disk or any directory under the shared disk drive (example: Y: or Y:\JP1).

The directory name cannot contain any spaces.

The directory NNM is created immediately below the specified path (example: $Y : \NNM$ or $Y : \JP1 \NNM$).

The storage directory cannot be renamed.

(c) Configuring NNMi for HA

Next, run NNMi's HA configuration.

1. Create an NNMi HA resource group:

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

For details about the configuration items for this command, see 19.9.2 NNMi-provided HA configuration scripts. Make sure that you specify disk, not none, for the shared disk type. Specify for the shared disk the path specified in step 3 in subsection (b).

Configuration example

The HA configuration items are listed below in the order they are entered interactively to nnmhaconfigure.ovpl. Enter appropriate values based on the information provided in Table 19-3: NNMi HA primary cluster node configuration information in 19.4.2 Configuring NNMi for HA.

HA configuration item	Example setting
HA resource group name	jp1ha1
Virtual host name	lhost1

^{19.} Configuring NNMi in a High Availability Cluster

HA configuration item	Example setting
Network interface of the virtual host	Local area connection
Type of shared file system	disk (make sure that you specify disk)
Directory to be mounted	Y drive

Important

Before you execute the configuration command, check the following notes:

- If a value specified in nnmhaconfigure.ovpl is already in use by another resource group or resource, a resource creation error occurs. Before you execute nnmhaconfigure.ovpl, verify that the specified values are not already in use.
- If a specified resource group name, IP address, or disk is already in use, the cluster software command executed to create resources results in an error. If an error occurs, nnmhaconfigure.ovpl terminates abnormally at that point, in which case the resource group and resources that had been created up to that point remain. Delete those remaining resources before you resolve the error and re-execute nnmhaconfigure.ovpl.
- For the network interface for which a virtual address is set, verify the following:
- In **Networks** in the Failover Cluster Management console, verify the resources that include a network address of logical IP address.

Execution example

The following shows an example screen display in which the example configuration values are specified, where each input item follows a question mark (?).

```
C:\Program Files (x86)\Hitachi\Cm2NNMi\misc\nnm\ha>nnmhaconfigure.ovpl NNM
QUESTION: Enter the name of HA resource group: ? jplhal
A primary node configuration has been discovered.
QUESTION: Enter a valid virtual host name: ? lhost1
Available network interface:
Network subnet mask Network interface
255.255.255.0 Cluster network 3
255.255.255.0
                     Cluster network 1
Available value:
1: Cluster network 3
2: Cluster network 1
QUESTION: Enter the type of shared file system: ? 2
Available value:
1: disk
2: none
QUESTION: Enter the type of shared file system (disk, none): ? 1
QUESTION: Enter the directory to mount disk: ? Y:
Creating a resource group.
Creating the resource group 'jplhal'...
                Node
Group
                             Status
----- -----
jp1ha1
                  NNMX64-33
                                 Offline
```

^{19.} Configuring NNMi in a High Availability Cluster

Creating the resource group 'lhost1'... Resource Status Group Node _____ jp1ha1 lhost1 NNMX64-33 Offline Making the resource 'lhost1' dependent on the resource 'jplhal-IP'... Configuring the HA value C:/ProgramData/Hitachi/Cm2NNMi/shared/nnm/conf/ov.conf. Disabling the automatic startup of HP OpenView Process Manager service. [SC] ChangeServiceConfig SUCCESS Note: Updating NNMi FQDN to match the specified virtual host name. Configuring fqd n to lhost1.xxx.xxx. Configuring the domain to xxx.xxx. Microsoft (R) Windows Script Host Version 5.7 Copyright (C) Microsoft Corporation 1996-2001. All rights reserved. Generating a new SSL certificate. Generating a key store certificate of lhost1.xxx.xxx.selfsigned. [Completed successfully] Exporting the generated certificate to the trust store. The certificate has been saved in temporary.cert. The certificate has been added to the key store. C:\Program Files (x86) \ Hitachi\Cm2NNMi\misc\nnm\ha

2. Configure *resource-group* in such a manner that failover occurs in the event of a monitoring process error.

Open the properties of *resource-group*-APP, and then click the **Policies** tab.

Verify that **If restart is unsuccessful, fail over all resources in this service or application** is selected, and then set **Maximum restarts in the specified period** to 0.

If it is selected, clear the check box If all the restart attempts fail, begin restarting again after the specified period (*hh:mm*).

Important

Configuration of *resource-group* and the resources registered to *resource_group* is used to specify actions such as error handling. For details about the role of each configuration item, see the cluster service Help.

3. On the primary cluster node, disable the NNMTrapReceiver service automatic start feature.

From the Start menu, select Administrative Tools > Services, and select NNM Trap Receiver and NNM Trap Receiver Manager, and then set Startup type to Manual.

4. By restarting, the specified configuration settings are applied and the NNMi environment variables are loaded. You can start and stop the service by executing the net start ClusSvc and net stop ClusSvc commands, respectively.

Important

If you use HTTPS communications to access the NNMi server, you must configure the cluster to use an appropriate certificate. For details, see 10.3.6 Working with Certificates in High-Availability Environments.

^{19.} Configuring NNMi in a High Availability Cluster

(d) Verifying the startup

Lastly, verify the startup.

1. Start the NNMi HA resource group.

Execute the start command on the primary cluster node.

• Execute the following start command:

%NnmInstallDir%misc\nnm\ha\nnmhastartrg.ovpl NNM resource-group

• Verify that *resource-group* has started. If NNMi does not start successfully, see 19.8 Troubleshooting the HA Configuration.

NNMi is now running under HA.

Important

Do not use the ovstart and ovstop commands for normal NNMi operation in the HA configuration. Use these commands only when instructed to do so for HA maintenance purposes. To start and stop NNMi in the HA configuration, start or stop the HA resource group by using the cluster software.

(3) Configuring NNMi on the secondary cluster node

(a) Preparations

Start with the preparations:

- 1. If you have not already done so, complete the procedure described in 19.2 Verifying the prerequisites to configuring NNMi for HA
- 2. If you have not already done so, install NNMi, and then verify that NNMi is working correctly.
- 3. Restore the backup data.

Restore onto the secondary cluster node the backup data obtained in step 3 in subsection (a) in (2) Configuring NNMi on the primary cluster node.

%NnmInstallDir%bin\nnmrestore.ovpl -force -partial -source backup-data

For details about this command, see 20. NNMi Backup and Restore Tools.

(b) Configuring NNMi for HA

Next, run NNMi's HA configuration.

1. Stop NNMi.

```
%NnmInstallDir%bin\ovstop -c
net stop NnmTrapReceiver
```

2. Configure the NNMi HA resource group:

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

```
Specify the HA resource group name when prompted by the command.
```

Execution example

^{19.} Configuring NNMi in a High Availability Cluster

C:\Program Files (x86)\Hitachi\Cm2NNMi\misc\nnm\ha>nnmhaconfigure.ovpl NNM
QUESTION: Enter the name of HA resource group: ? jplhal
a secondary node configuration has been discovered.
Disabling the automatic startup of HP OpenView Process Manager service.
[SC] ChangeServiceConfig SUCCESS
Note: Updating NNMi FQDN to match the specified virtual host name. Configuring fqd
n to lhost1.xxx.xxx.
Configuring the domain to xxx.xxx.
Microsoft (R) Windows Script Host Version 5.7
Copyright (C) Microsoft Corporation 1996-2001. All rights reserved.
Generating a new SSL certificate.
C:\Program Files (x86)\ Hitachi\Cm2NNMi\misc\nnm\ha

3. On the secondary cluster node, disable the NNMTrapReceiver service automatic start feature.

From the Start menu, select Administrative Tools > Services, and select NNM Trap Receiver and NNM Trap Receiver Manager, and then set Startup type to Manual.

4. Verify that configuration was successful:

%NnmInstallDir%misc\nnm\ha\nnmhaclusterinfo.ovpl -group resource-group -nodes

This command outputs a list of all nodes that are in the specified HA resource group.

5. On the secondary cluster node, restart the cluster service.

By restarting, the specified configuration settings are applied and the NNMi environment variables are loaded. You can start and stop the service by executing the net start ClusSvc and net stop ClusSvc commands, respectively.

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

Important

Problems might occur in the created resource group, such as the service no longer starts successfully if the NNMi default values for resource group and resource configuration are changed.

Special care is needed if the following configuration item is changed to a value that is smaller than its default value:

• Length of period during which the cluster service waits until it restarts a resource

WSFC standard installation:

Period on the **Policies** tab for the *resource-group*-APP properties (default value: 30:00 minutes) DeadlockTimeout value for the *resource-group*-APP properties (default value: 2,700,000 milliseconds)

19.4.4 Configuring NNMi for HA (Linux)

This subsection explains how to configure NNMi for HA in a Linux environment.

^{19.} Configuring NNMi in a High Availability Cluster

In HA configuration for NNMi, you create a new resource group for NNMi. Therefore, you must begin the configuration procedure when there is no resource group to be configured.

The script (nnmhaconfigure.ovpl) used to configure NNMi for HA internally creates a resource group and individual resources for the cluster software. When the configuration procedure is completed, the following resource group has been configured.

Table 19-9: Components of resource group for NNMi in Veritas Cluster Server or Symantec Cluster Server

Resource name	Resource type	Description
<i>resource-group-</i> ip	IP	Controls virtual IP addresses.
<i>resource-group-</i> dg	DiskGroup	Controls disk groups.
<i>resource-group-</i> volume	Volume	Controls volumes.
resource-group-mount	Mount	Controls shared file systems.
resource-group-app	Application	Controls the start, stop, and monitoring of NNMi.

In VCS or SCS, nnmhaconfigure.ovpl configures the above resources by internally executing commands, such as hagrp and hares.

- *resource-group* is replaced with the HA resource group name.
- The resource dependency is set so that Volume requires DiskGroup and IP, Mount requires Volume, and Application requires Mount and IP.
- The resources for monitoring network interfaces by VCS or SCS (NIC of VCS or SCS) are not configured. If necessary, add configurations.
- If it takes too long for NNMi to start and a timeout results, tune the OnlineTimeout setting of the *resource-group*-app by referencing 19.8 Troubleshooting the HA Configuration.

The following shows an example of configuration for each resource.

Example: Definition of VCS or SCS configuration file main.cf

Angle brackets (< >) enclose the setting values specified in nnmhaconfigure.ovpl.

```
group <resource group> (
  SystemList = \{ < node1 > = 1 , < node2 > = 1 \}
 UserStrGlobal = "NNM INTERFACE=<virtual host>;HA LOCALE=<LOCALE>;HA MOUNT POINT=<mo
untpoint>"
 )
 Application <resource group>-app (
     StartProgram = "/opt/OV/misc/nnm/ha/nnmharg.ovpl NNM -start <resource group>"
     StopProgram = "/opt/OV/misc/nnm/ha/nnmharg.ovpl NNM -stop <resource_group>"
     CleanProgram = "/opt/OV/misc/nnm/ha/nnmharg.ovpl NNM -clean <resource group>"
     MonitorProgram = "/opt/OV/misc/nnm/ha/nnmharg.ovpl NNM -monitor
                       <resource group>"
     OnlineTimeout = 1800
  DiskGroup <resource group>-dg (
     DiskGroup = \langle disk group \rangle
  IP <resource group>-ip (
     Device = <network interface of virtual host>
     Address = "10.208.228.159"
```

^{19.} Configuring NNMi in a High Availability Cluster

```
NetMask = "255.255.255.0"
  )
Mount <resource group>-mount (
  MountPoint = "<mountpoint>"
   BlockDevice = "/dev/vx/dsk/<disk group>/<volume group>"
   FSType = <type_of_shared_file_systems>
   FsckOpt = "-y"
Volume <resource group>-volume (
   Volume = <volume group>
   DiskGroup = <disk group>
   )
<resource group>-app requires <resource group>-ip
<resource group>-app requires <resource group>-mount
<resource group>-mount requires <resource group>-volume
<resource group>-volume requires <resource group>-dg
<resource group>-volume requires <resource group>-ip
```

Table 19-10: Components of resource group for NNMi in HA Monitor

Configuration item	Setting (control script)
name (start)	/var/opt/0V/hacluster/ <i>resource-group</i> /cm2_start.sh
termcommand (stop)	/var/opt/0V/hacluster/ <i>resource-group</i> /cm2_stop.sh
patrolcommand (monitoring)	/var/opt/OV/hacluster/ <i>resource-group</i> /cm2_monitor.sh

• *resource-group* is replaced with the HA resource group name.

Important

For HA Monitor, configure NNMi without using nnmhaconfigure.ovpl. For details about the procedure, see the *Release Notes*.

(1) Configuring NNMi on the primary cluster node

Complete the procedure described below on the primary cluster node.

(a) Preparations

Start with the preparations:

- 1. If you have not already done so, complete the procedure described in 19.2 Verifying the prerequisites to configuring NNMi for HA
- 2. If you have not already done so, install NNMi, and then verify that NNMi is working correctly.
- 3. Use the following command to back up all NNMi settings and data:

Example:

/opt/OV/bin/nnmbackup.ovpl -scope all -target directory

For details about this command, see 20. NNMi Backup and Restore Tools.

In the initial status of NNMi cluster environment configuration, the data in the primary cluster node must exactly match the data in the secondary cluster node. Therefore, restore the backup data obtained here during the secondary cluster node configuration procedure.

^{19.} Configuring NNMi in a High Availability Cluster

(b) Copying data to the shared disk

Next, copy data to the shared disk.

- 1. Create the directory mount point for the shared disk.
- 2. Provide a shared disk for the NNMi HA resource group.

Important

Verify that the provided shared disk satisfies the following conditions:

- It has already been formatted
- It has enough free space
- It is not being used by any other resource group
- 3. Activate the shared disk and then mount.

Example:

• Using VxVM/VxFS for disk management in Linux VCS or SCS

```
vxdg import disk-group
vxvol -g disk-group startall
mount -t vxfs /dev/vx/dsk/disk-group/volume HA-mount-point
```

Verify that the shared disk directory mount point has been created with root as the user, root as the group, and the permissions set to 755.

Example:

ls -l

4. Stop NNMi:

· For distribution that manages services with systemd

```
/opt/OV/bin/ovstop -c
/opt/OV/bin/nettrap stop
```

• For other distribution:

```
/opt/OV/bin/ovstop -c
/etc/init.d/nettrap stop
```

5. Copy the NNMi files to the shared disk:

```
/opt/OV/misc/nnm/ha/nnmhadisk.ovpl NNM -to HA-mount-point
```

Important

The directory NNM is created immediately below the specified mount point (HA-mount-point/NNM).

The storage directory cannot be renamed.

6. Unmount the shared disk and deactivate it.

Example:

 Configuration using VCS or SCS and VxVM/VxFS umount *HA-mount-point*

^{19.} Configuring NNMi in a High Availability Cluster

vxvol -g disk-group stopall
vxdg deport disk-group

(c) Configuring NNMi for HA

Next, run NNMi's HA configuration.

1. Create an NNMi HA resource group:

/opt/OV/misc/nnm/ha/nnmhaconfigure.ovpl NNM

For details about the configuration items for this command, see 19.9.2 NNMi-provided HA configuration scripts. For the shared disk type, make sure that you specify disk, not none.

Configuration example

The HA configuration items are listed below in the order they are entered interactively to nnmhaconfigure.ovpl. Enter appropriate values based on the information provided in Table 19-3: NNMi HA primary cluster node configuration information in 19.4.2 Configuring NNMi for HA.

HA configuration item	Example setting
HA resource group name	jplhal
Virtual host name	lhostl
Network interface of the virtual host	lan0
Type of shared file system	disk (make sure that you specify disk)
Disk type	vxfs
Disk group (VCS or SCS only)	shdg3
Volume group	vg03
Directory to be mounted	/shdsk1

Important

Before you execute the configuration command, check the following notes:

• NNMi in an HA configuration starts by using the nnmhaconfigure.ovpl runtime locale. Verify that the window used during the execution of nnmhaconfigure.ovpl is set to the correct locale (LANG environment variable):

Linux VCS or Linux SCS: C

To change the locale after HA configuration, see 19.6 Maintaining the HA Configuration.

- If a value specified in nnmhaconfigure.ovpl is already in use by another resource group or resource, a resource creation error occurs. Before you execute nnmhaconfigure.ovpl, verify that the specified values are not already in use.
- If a specified resource group name, IP address, or disk is already in use, the cluster software command executed to create resources results in an error. If an error occurs, nnmhaconfigure.ovpl terminates abnormally at that point, in which case the resource group and resources that had been created up to that point remain. Delete those remaining resources before you resolve the error and re-execute nnmhaconfigure.ovpl.
- While nnmhaconfigure.ovpl is running, the message shown below might be displayed. This is a message for internal processing and no action is needed.

19. Configuring NNMi in a High Availability Cluster
The disk group was not found. Import will be attempted. Unable to perform the security token exchange with cmclconfd on node *xxxxx* Cannot connect to configuration daemon (cmclconfd) on node *xxxxx*

Execution example

The following shows an example screen display in which the example configuration values are specified, where each input item follows a question mark (?).

• Example for VCS or SCS (Linux)

```
# /opt/OV/misc/nnm/ha/nnmhaconfigure.ovpl NNM
QUESTION: Enter the name of HA resource group:
                                                ? jplhal
A primary node configuration has been discovered.
QUESTION: Enter a valid virtual host name: ? lhost1
Information: Use of network interface information:
Network interface: bond0
Network subnet mask: 255.255.255.0
Available value:
1: disk
2: none
QUESTION: Enter the type of shared file system (disk, none): ? 1
Available value:
1: vxfs
2: ext2
3: ext3
QUESTION: Enter the name of disk type: ? 1
QUESTION: Enter the name of disk group: ? shdg3
The disk group was not found. Import will be attempted.
QUESTION: Enter the name of volume group: ? shvol3
QUESTION: Enter the directory to mount disk: ? /shdsk1
Creating a resource group.
VCS NOTICE V-16-1-10136 Group added; populating SystemList and setting the Paralle
1 attribute recommended before adding resources
Configuring the HA value /var/opt/OV/shared/nnm/conf/ov.conf.
Deleting the boot script.
Note: Updating NNMi FQDN to match the specified virtual host name. Configuring fqd
n to lhost1.
Configuring the domain to xxx.xxx.
Generating a new SSL certificate.
lhost1.xxx.xxx.selfsigned.
[Completed successfully]
Exporting the generated certificate to the trust store.
The certificate has been saved in temporary.cert.
The certificate has been added to the key store.
```

- 2. On the primary cluster node, disable the service automatic start feature. Use the following commands to disable the service automatic start feature:
 - RHEL 6

^{19.} Configuring NNMi in a High Availability Cluster

unlink/etc/rc0.d/K01nettrap unlink/etc/rc1.d/K01nettrap unlink/etc/rc2.d/K01nettrap unlink/etc/rc3.d/S98nettrap unlink/etc/rc5.d/S98nettrap unlink/etc/rc6.d/K01nettrap

• Other distributions

systemctl disable netmgt.service systemctl disable nettrap.service systemctl stop netmgt.service systemctl stop nettrap.service

3. In VCS or SCS, enable the created resources (set Enabled to 1).

Example:

hares -modify *resource-group*-app Enabled 1 hares -modify *resource-group*-dg Enabled 1

hares -modify *resource-group*-ip Enabled 1

hares -modify *resource-group*-mount Enabled 1

hares -modify *resource-group*-volume Enabled 1

Next, set the VCS or SCS settings to read-only, and then output the VCS or SCS configuration file main.cf:

haconf -dump -makero

The resources for monitoring network interfaces by VCS or SCS (such as NIC, MultiNICA, and MultiNICB of VCS or SCS) are not configured. If necessary, add configurations.

Important

If you use HTTPS communications to access the NNMi server, you must configure the cluster to use an appropriate certificate. For details, see 10.3.6 Working with Certificates in High-Availability Environments.

(d) Verifying the startup

Lastly, verify the startup.

1. Start the NNMi HA resource group.

/opt/OV/misc/nnm/ha/nnmhastartrg.ovpl NNM resource-group

This command returns the prompt with waiting for the start of the HA resource group. Use the command of HA cluster software to verify that the resource group has started.

If NNMi does not start successfully, see 19.8 Troubleshooting the HA Configuration.

NNMi is now running under HA.

^{19.} Configuring NNMi in a High Availability Cluster

Important

Do not use the ovstart and ovstop commands for normal NNMi operation in the HA configuration. Use these commands only when instructed to do so for HA maintenance purposes. To start and stop NNMi in the HA configuration, start or stop the HA resource group by using the cluster software.

(2) Configuring NNMi on the secondary cluster node

(a) Preparations

Start with the preparations:

- 1. If you have not already done so, complete the procedure described in 19.2 Verifying the prerequisites to configuring NNMi for HA.
- 2. If you have not already done so, install NNMi, and then verify that NNMi is working correctly.
- 3. Restore the backup data.

Restore onto the secondary cluster node the backup data obtained in step 3 in subsection (a) in (1) Configuring NNMi on the primary cluster node.

/opt/OV/bin/nnmrestore.ovpl -force -partial -source backup-data

For details about this command, see 20. NNMi Backup and Restore Tools.

(b) Configuring NNMi for HA

Next, run NNMi's HA configuration.

1. Create the mount point for the shared disk.

You must use for this mount point the same name as used for the mount point created in step 1 in subsection (b) in (1) Configuring NNMi on the primary cluster node.

- 2. Stop NNMi.
 - For distribution that manages services with systemd

```
/opt/OV/bin/ovstop -c
/opt/OV/bin/nettrap stop
```

• For other distribution:

```
/opt/OV/bin/ovstop -c
/etc/init.d/nettrap stop
```

3. Configure the NNMi HA resource group:

/opt/OV/misc/nnm/ha/nnmhaconfigure.ovpl NNM

Specify the HA resource group name when prompted by the command.

Execution example

```
# /opt/OV/misc/nnm/ha/nnmhaconfigure.ovpl NNM
QUESTION: Enter the name of HA resource group: ? jplhal
A secondary node configuration has been discovered.
Completed the cluster update
Deleting the boot script.
Note: Updating NNMi FQDN to match the specified virtual host name. Configuring fqd
```

^{19.} Configuring NNMi in a High Availability Cluster

```
n to lhost1.xxx.xxx.
Setting the domain to .xxx.xxx.
Generating a new SSL certificate.
#
```

4. On the secondary cluster node, disable the service automatic start feature. Use the following commands to disable the service automatic start feature:

• RHEL 6

unlink/etc/rc0.d/K01nettrap unlink/etc/rc1.d/K01nettrap unlink/etc/rc2.d/K01nettrap unlink/etc/rc3.d/S98nettrap unlink/etc/rc5.d/S98nettrap unlink/etc/rc6.d/K01nettrap

• Other distributions

systemctl disable netmgt.service systemctl disable nettrap.service systemctl stop netmgt.service systemctl stop nettrap.service

5. In VCS or SCS, apply configuration changes to the HA cluster:

haconf -dump -makero

6. Verify that configuration was successful:

/opt/OV/misc/nnm/ha/nnmhaclusterinfo.ovpl -group resource-group -nodes

This command outputs a list of all nodes that are in the specified HA resource group.

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

19.5 Shared NNMi Data

Because NNMi running in an HA environment shares files among NNMi nodes in the HA cluster, you must use a shared disk that is accessible from all NNMi nodes and that is controlled by the HA product.

Important

A configuration that uses NFS or CIFS connection for shared disk is not supported.

19.5.1 Data on the NNMi shared disk

This subsection 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:

• In Windows

- %NnmDataDir% maps to %HA MOUNT POINT%\NNM\dataDir

• In Linux

- \$NnmDataDir maps to \$HA_MOUNT_POINT/NNM/dataDir

The following are the main directories that are moved to the shared disk:

• In Windows

```
- %NnmDataDir%shared\nnm\databases\Postgres
```

Embedded database

- NnmDataDir%log\nnm
- NNMi logging directory
- -%NnmDataDir%shared\nnm\databases\custompoller

Custom poller collections export directory

-%NnmDataDir%nmsas\NNM\log

NNMi audit log directory

- -%NnmDataDir%nmsas\NNM\conf
- NNMi directory for configuring audit log files
- -%NnmDataDir%nmsas\NNM\data
- Transaction store used by ovjboss
- In Linux
 - \$NnmDataDir/shared/nnm/databases/Postgres

Embedded database

- \$NnmDataDir/log/nnm

NNMi logging directory

- \$NnmDataDir/shared/nnm/databases/custompoller

- Custom poller collections export directory
- \$NnmDataDir/nmsas/NNM/log

NNMi audit log directory

^{19.} Configuring NNMi in a High Availability Cluster

\$NnmDataDir/nmsas/NNM/conf
NNMi directory for configuring audit log files
\$NnmDataDir/nmsas/NNM/data
Transaction store used by ovjboss

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

19.5.2 Replication of configuration files

NNMi running in an HA environment 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 monitors the updating of configuration files located on the local disk. When a configuration file is updated, the command copies it to the shared disk. In the event of a failover, the command copies the most recent configuration files from the shared disk to the target node. After that, NNMi is started.

The above update verification and copying of the configuration files are a part of the NNMi monitoring process that is run periodically from the HA cluster. Therefore, if node switchover occurs after configuration files have been changed but before they were copied, the changes are not applied. In such a case, you must change the configuration again.

In the nnmdatareplicator.conf file, specify the NNMi folders and files to be included in data replication.

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

^{19.} Configuring NNMi in a High Availability Cluster

19.6.1 Placing NNMi in maintenance mode

The maintenance mode is a function for temporarily disabling failover during NNMi maintenance.

NNMi running in an HA environment is monitored by the HA product. If NNMi stops, the HA product determines that a failure has occurred and fails over from the primary node to the secondary node. This means that failover also occurs when NNMi is stopped for maintenance purposes.

The maintenance mode disables failover processing by suppressing the monitoring of NNMi. This means that you can perform maintenance work by executing ovstop and ovstart on the active cluster node. Make sure that you do not execute ovstop and ovstart on the passive cluster node.

Important

If a product that requires NNMi is running, placing only NNMi in the maintenance mode will still trigger failover if the related product fails. In such a case, first stop the related product or place it in the mode equivalent to the maintenance mode, and then place NNMi in the maintenance mode.

(1) Placing NNMi in maintenance mode

When NNMi is placed in the maintenance mode, NNMi's monitoring process is disabled. While NNMi is in the maintenance mode, stopping or starting NNMi for that HA resource group does not trigger failover.

To place NNMi in the maintenance mode, create the following file (which can be empty) on the active cluster node:

• Windows

%NnmDataDir%hacluster\resource-group\maintenance

• Linux

\$NnmDataDir/hacluster/resource-group/maintenance

(2) Removing NNMi from maintenance mode

Taking NNMi out of maintenance mode re-enables NNMi monitoring. If NNMi is stopped, the HA resource group will fail over to the passive cluster node.

To remove an HA resource group from maintenance mode:

1. Verify that NNMi is running correctly:

```
ovstatus -c
```

All NNMi services must show the state RUNNING.

2. Delete the maintenance file from the node that was the active cluster node before maintenance was initiated. For details about the maintenance file, see (1) Placing NNMi in maintenance mode.

^{19.} Configuring NNMi in a High Availability Cluster

19.6.2 Maintaining NNMi in an HA cluster

Note

For HA Monitor, some of the procedures might differ. For the actual procedures, see the Release Notes.

(1) 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 (nnmhastartrg.ovpl and nnmhastoprg.ovpl) or the appropriate HA product commands for starting and stopping HA resource groups.

(2) Changing NNMi host names and IP addresses in a cluster environment

(a) Renaming virtual hosts

NNMi's virtual hosts cannot be renamed after NNMi has been configured for HA. To rename a virtual host, unconfigure NNMi for HA and then re-configure NNMi for HA under the new virtual host name.

(b) Changing virtual IP addresses

To change the virtual IP address of the NNMi HA resource group, perform the following steps on the active cluster node:

1. Stop the NNMi HA resource group:

• Windows

%NnmInstallDir%misc\nnm\ha\nnmhastoprg.ovpl NNM resource-group

net stop NnmTrapReceiver

• Linux

- For distributions that uses systemd to manage services

/opt/OV/misc/nnm/ha/nnmhastoprg.ovpl NNM resource-group

/opt/OV/bin/nettrap stop

- Other distributions

/opt/OV/misc/nnm/ha/nnmhastoprg.ovpl NNM resource-group

/etc/init.d/nettrap stop

- 2. Change the cluster configuration to use the new IP address:
 - Windows

In the cluster's management console, change the IP address resource configuration. Open the resource group, double-click *resource-group-ip*, select parameters, and then enter the new IP address.

• Linux

/opt/OV/misc/nnm/ha/nnmhargconfigure.ovpl NNM resource-group -set_value resourcegroup-ip Address new-IP-address

Run haconf -dump -makero to apply the configuration change to the HA cluster.

- 3. Start the NNMi HA resource group:
 - Windows

^{19.} Configuring NNMi in a High Availability Cluster

%NnmInstallDir%misc\nnm\ha\nnmhastartrg.ovpl NNM resource-group

• Linux

/opt/OV/misc/nnm/ha/nnmhastartrg.ovpl NNM resource-group

- 4. Verify that NNMi started correctly:
 - Windows

```
%NnmInstallDir%bin\ovstatus-c
```

• Linux /opt/OV/bin/ovstatus-c

(c) Renaming physical hosts

If you need to rename a physical host in the system in a cluster environment, do so after stopping NNMi. No NNMi settings are required. After the physical host has been renamed, start NNMi.

Note that physical host names (computer names) cannot be changed in Windows cluster environments.

(d) Changing physical IP addresses

For details about the procedure for changing a physical IP address, see 22.4 Changing the IP address of a stand-alone NNMi management server.

(3) Stopping NNMi without causing failover

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

- 1. If there are related products that require NNMi, stop those products or place them into a mode equivalent to maintenance mode.
- 2. Put the HA resource group in maintenance mode as described in (1) Placing NNMi in maintenance mode.
- 3. Stop NNMi:

```
ovstop -c
```

(4) 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 must show the state RUNNING.

- 3. Take the HA resource group out of maintenance mode as described in (2) Removing NNMi from maintenance mode.
- 4. If you have stopped or placed NNMi-related products into maintenance mode, restore them to their initial state.

^{19.} Configuring NNMi in a High Availability Cluster

(5) Backing up NNMi configured for HA

(a) Online backup

If you use online backups, verify that the shared disk is accessible from the active cluster node, and then perform the normal backup procedure.

(b) Offline backup

To make an offline backup of NNMi configured for HA, perform the procedure described below. For details about the maintenance mode indicated in the procedure, see 19.6.1 Placing NNMi in maintenance mode.

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

```
%NnmInstallDir%misc\nnm\ha\nnmhaclusterinfo.ovpl-groupresource-group-activeNode
```

• Linux

```
/opt/OV/misc/nnm/ha/nnmhaclusterinfo.ovpl-group resource-group -activeNode
```

- 2. Place the active cluster node in the maintenance mode.
- 3. Stop NNMi:

ovstop -c

- 4. Verify that the shared disk is online by using an HA product operation. If the shared disk is offline, place it online.
- 5. Verify that the shared disk is accessible, and then execute the nnmbackup.ovpl command to perform an offline backup and obtain backup data.
- 6. Start NNMi:

ovstart -c

Wait until NNMi completes its startup.

7. After NNMi services have started, release maintenance mode.

(6) Restoring NNMi configured for HA

To restore backup data, perform the procedure described below. For details about the maintenance mode indicated in the procedure, see 19.6.1 Placing NNMi in maintenance mode.

Important

Do not restore backup data obtained from an NNMi with a single configuration to an NNMi with a cluster configuration.

1. Restore NNMi to a status in which NNMi runs successfully as a cluster and is configured for HA.

For example, if part or all of the system has failed due to a hardware failure, restore the system so that NNMi runs with the HA configuration.

- 2. Set the node used to perform restore processing as the active cluster node.
- 3. Place the active cluster node in the maintenance mode.
- 4. Perform restore processing.

^{19.} Configuring NNMi in a High Availability Cluster

- For backup data obtained by using the nnmbackup.ovpl command Use the nnmrestore.ovpl command to restore the backup data.
- For backup data obtained by using the nnmbackupembdb.ovpl command

Use the nnmrestoreembdb.ovpl command to restore the backup data.

Important

If you use the nnmrestore.ovpl command to restore backup data obtained on another node, do not specify the lic option so that that node's license is not applied.

5. Start NNMi:

```
ovstart -c
```

- 6. Release maintenance mode.
- 7. Perform steps 2 through 6 on the remaining node.

These steps are necessary in order to synchronize the configuration files on all nodes' local disks. Make sure that you restore the same backup data.

If you use the nnmrestoreembdb.ovpl command, restore the backup data only on one node because this command restores a database on the shared disk.

(7) Initializing the database

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

• Windows

```
%NnmInstallDir%misc\nnm\ha\nnmhaclusterinfo.ovpl-group resource-group -activeNode
```

• Linux

/opt/OV/misc/nnm/ha/nnmhaclusterinfo.ovpl -group resource-group -activeNode

- 2. Place the active cluster node in the maintenance mode.
- 3. Verify that the shared disk is accessible.
- 4. Initialize the database by executing the nnmresetembdb.ovpl command with no argument specified:
 - Windows

%NnmInstallDir%bin\nnmresetembdb.ovpl

• Linux

```
/opt/OV/bin/nnmresetembdb.ovpl
```

- 5. Execute ovstatus -c to verify that the NNMi services are running.
- 6. Release the maintenance mode.

^{19.} Configuring NNMi in a High Availability Cluster

19.7 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 stand-alone management server, or you can uninstall NNMi from that node.

To completely unconfigure NNMi from an HA cluster, follow the steps described below.

- 19.7.1 Determining the active cluster node
- 19.7.2 Unconfiguring NNMi on the passive cluster node
- 19.7.3 Unconfiguring NNMi on the active cluster node

To unconfigure NNMi on the active cluster node, the procedure for deleting the NNMi data is described as well as the procedure for using the NNMi data on the single server after NNMi has been unconfigured from the HA cluster.

🖹 Note

For HA Monitor, unconfigure NNMi without using nnmhaunconfigure.ovpl. For details about the procedure, see the *Release Notes*.

19.7.1 Determining the active cluster node

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

```
%NnmInstallDir%misc\nnm\ha\nnmhaclusterinfo.ovpl-group resource-group -activeNode
```

Linux

/opt/OV/misc/nnm/ha/nnmhaclusterinfo.ovpl-group resource-group -activeNode

19.7.2 Unconfiguring NNMi on the passive cluster node

- 1. On the passive cluster node, unconfigure NNMi from the HA cluster:
 - Windows

```
net stop NnmTrapReceiver
```

%NnmInstallDir%misc\nnm\ha\nnmhaunconfigure.ovpl NNM resource-group

• Linux

- For distributions that uses systemd to manage services

/opt/OV/bin/nettrap stop

/opt/OV/misc/nnm/ha/nnmhaunconfigure.ovpl NNM resource-group

- Other distributions

/etc/init.d/nettrap stop

/opt/OV/misc/nnm/ha/nnmhaunconfigure.ovpl NNM resource-group

This command removes the corresponding node from the list of nodes in the HA resource group. It does not change the settings for running NNMi configured for HA or data on the shared disk of any other node.

^{19.} Configuring NNMi in a High Availability Cluster

The following message might be displayed, but this is not a problem:

• Windows

```
Warning: There is no value in the public entry PUBLIC.HA_MOUNT_POINT in the resource group xxxxx in the cluster registry.
```

Important

In a VCS or SCS environment where NNMi is in an HA configuration and the physical host name has been changed, before the HA configuration is canceled, the physical host name must be reverted to the name that was used when HA was configured. To continue using the new host name even after switching to a single server configuration, cancel the HA configuration, and then configure the required settings as described in 22.5 Changing the host name or domain name of an NNMi management server.

2. In VCS or SCS, apply the changes to the HA cluster.

```
haconf -dump -makero
```

3. Change the FQDN setting for the NNMi node to the physical host name.

a. Edit the nms-local.properties file.

File path

```
- Windows: %NnmDataDir%conf\nnm\props\nms-local.properties
```

```
- Linux: /var/opt/OV/conf/nnm/props/nms-local.properties
```

What to edit

com.hp.ov.nms.fqdn = virtual-host-name

Specify a virtual host name, not a physical host name.

b. Execute the nmsetofficialfqdn.ovpl command.

For FQDN, specify the FQDN of the physical host name (host name displayed by the hostname command).

• Windows

%NnmInstallDir%bin\nnmsetofficialfqdn.ovpl-force FQDN

• Linux

```
/opt/OV/bin/nnmsetofficialfqdn.ovpl-force FQDN
```

This command changes the FQDN setting that was changed to the virtual host name during HA configuration to the FQDN of the physical host name.

The following messages might be displayed during command execution:

• For single sign-on to function normally, you must generate a new certificate manfully

Ignore this message because single sign-on is not supported.

• A new certificate cannot be generated. The self-signed alias xxx.xxx already exists in the key store.

If it is displayed, rename the keystore backup file from nnm-key.p12.xxxxxxxxxx to nnm-key.p12, and rename the truststore backup file from nnm-trust.p12.xxxxxxxxxx to nnm-trust.p12 in the following directory.

- Windows %NnmDataDir%shared\nnm\certificates\
- Linux /var/opt/OV/shared/nnm/certificates/

19. Configuring NNMi in a High Availability Cluster

If multiple backup files exist, rename the file whose file name ends in the largest 14-digit number.

As a result, a self-signed certificate is used. If you want to use a different self-signed certificate or a Certificate Authority (CA)-signed certificate for NNMi communications, you must do some additional work. For details about the certificates, see 10. Working with Certificates for NNMi.

4. Move the NNMi HA resource group-specific files to a separate location for safekeeping.

If you do not plan to reconfigure the NNMi HA resource group, you do not need to save a copy of these files, and you can delete them at this time.

• Windows

In Explorer, delete the %NnmDataDir%hacluster\resource-group\ folder.

• Linux

```
cd/var/opt/OV/hacluster/
```

- rm -r resource-group
- 5. Delete the following file:

File path

- Windows

```
In Explorer, delete %NnmDataDir%shared\nnm\databases\nnmdatareplicator \DataReplicator.db.
```

- Linux

```
rm /var/opt/OV/shared/nnm/databases/nnmdatareplicator/DataReplicator.db
```

- 6. Enable the service automatic start feature.
 - Windows

From the Start menu, select Administrative Tools > Services, and select NNM Trap Receiver and NNM Trap Receiver Manager, and then set Startup type to Automatic.

• RHEL6

```
ln -s /etc/init.d/nettrap /etc/rc0.d/K01nettrap
ln -s /etc/init.d/nettrap /etc/rc1.d/K01nettrap
ln -s /etc/init.d/nettrap /etc/rc2.d/K01nettrap
ln -s /etc/init.d/nettrap /etc/rc3.d/S98nettrap
ln -s /etc/init.d/nettrap /etc/rc5.d/S98nettrap
ln -s /etc/init.d/nettrap /etc/rc6.d/K01nettrap
```

• Other Linux distributions

```
systemctl enable netmgt.service
```

systemctl enable nettrap.service

systemctl start netmgt.service

systemctl start nettrap.service

The NNMi services are started when the systemctl start command is executed.

The procedure is complete.

If you want to remove NNMi completely from the HA cluster, unconfigure NNMi on the passive cluster nodes, and then unconfigure NNMi on the active cluster node.

^{19.} Configuring NNMi in a High Availability Cluster

19.7.3 Unconfiguring NNMi on the active cluster node

1. On the active cluster node, stop the NNMi HA resource group and NnmTrapReceiver.

• Windows

```
%NnmInstallDir%misc\nnm\ha\nnmhastoprg.ovpl NNM resource-group net stop NnmTrapReceiver
```

• Linux

- For distributions that uses systemd to manage services

```
$NnmInstallDir/misc/nnm/ha/nnmhastoprg.ovpl NNM resource-group
```

/opt/OV/bin/nettrap stop

/opt/OV/misc/nnm/ha/nnmhaunconfigure.ovpl NNM resource-group

- Other distributions

```
$NnmInstallDir/misc/nnm/ha/nnmhastoprg.ovpl NNM resource-group
/etc/init.d/nettrap stop
```

Important

In a VCS or SCS environment where NNMi is in an HA configuration and the physical host name has been changed, before a resource group is stopped, the physical host name must be reverted to the name that was used when HA was configured. To continue using the new host name even after switching to a single server configuration, cancel the HA configuration, and then configure the required settings as described in 22.5 Changing the host name or domain name of an NNMi management server.

2. On the active cluster node, unconfigure NNMi from the HA cluster:

• Windows

%NnmInstallDir%misc\nnm\ha\nnmhaunconfigure.ovpl NNM resource-group

• Linux

\$NnmInstallDir/misc/nnm/ha/nnmhaunconfigure.ovpl NNM resource-group

This command removes the corresponding node from the list of failover targets in the HA resource group. This command removes access to the shared disk but does not unconfigure the disk group or the volume group. The following message might be displayed, but this is not a problem:

• WSFC

Warning: There is no value in the public entry PUBLIC.HA_MOUNT_POINT in the resource group xxxxx in the cluster registry.

• VCS or SCS

VCS WARNING V-16-1-10133 Group does not exist: resource-group

3. Change the FQDN setting for the NNMi node to the physical host name.

a. Edit the nms-local.properties file.

File path

- Windows: %NnmDataDir%conf\nnm\props\nms-local.properties
- Linux: /var/opt/OV/conf/nnm/props/nms-local.properties

What to edit

com.hp.ov.nms.fqdn = virtual-host-name

```
19. Configuring NNMi in a High Availability Cluster
```

Specify a virtual host name, not a physical host name.

b. Execute the nnmsetofficialfqdn.ovpl command.

For FQDN, specify the FQDN of the physical host name (host name displayed by the hostname command).

- Windows %NnmInstallDir%bin\nnmsetofficialfqdn.ovpl-force FQDN
- Linux

/opt/OV/bin/nnmsetofficialfqdn.ovpl-force FQDN

This command changes the FQDN setting that was changed to the virtual host name during HA configuration to the FQDN of the physical host name.

The following messages might be displayed during command execution:

• For single sign-on to function normally, you must generate a new certificate manfully

Ignore this message because single sign-on is not supported.

• A new certificate cannot be generated. The self-signed alias xxx.xxx already exists in the key store.

If it is displayed, rename the keystore backup file from nnm-key.p12.xxxxxxxxx to nnm-key.p12, and rename the truststore backup file from nnm-trust.p12.xxxxxxxxx to nnm-trust.p12 in the following directory.

- Windows %NnmDataDir%shared\nnm\certificates\
- Linux /var/opt/OV/shared/nnm/certificates/

If multiple backup files exist, rename the file whose file name ends in the largest 14-digit number.

As a result, a self-signed certificate is used. If you want to use a different self-signed certificate or a Certificate Authority (CA)-signed certificate for NNMi communications, you must do some additional work. For details about the certificates, see 10. Working with Certificates for NNMi.

4. On the active cluster node, move the files that are specific to the NNMi HA resource group to a separate location for safekeeping.

If you do not plan to reconfigure the NNMi HA resource group, you do not need to save a copy of these files, and you can delete them at this time.

• Windows

In Explorer, delete the %NnmDataDir%hacluster\resource-group\ folder.

• Linux

cd/var/opt/OV/hacluster

rm -r resource-group

5. Delete the following file:

File path

- Windows

In Explorer, delete %NnmDataDir%shared\nnm\databases\nnmdatareplicator \DataReplicator.db.

- Linux

rm /var/opt/OV/shared/nnm/databases/nnmdatareplicator/DataReplicator.db

^{19.} Configuring NNMi in a High Availability Cluster

6. Mount the shared disk.

Use an OS or cluster operation to make the shared disk accessible.

Example:

In the **Server Manager**, please open the management screen of **Storage** Services > Disks, right-click on the disk where the shared disk has mounted and select **Onine**.

7. Copy the NNMi files from the shared disk to the previous active cluster node.

Perform this procedure if either of the following is applicable:

- You will be running NNMi with a database migrated from an HA configuration to a single-server configuration.
- You used nnmchangeembdbpw.ovpl in the HA configuration to change the database password.

Use the following command to copy the NNMi files from the shared disk to the local disk on the previous active cluster node:

• Windows

```
%NnmInstallDir%misc\nnm\ha\nnmhadisk.ovpl NNM -from HA-mount-point
```

• Linux

/opt/OV/misc/nnm/ha/nnmhadisk.ovpl NNM - from HA-mount-point

- 8. Delete the NNM folder or the NNM directory from the shared disk.
- 9. Unmount the shared disk.

Example:

In the Server Manager, please open the management screen of Storage Services > Disks, right-click on the disk where the shared disk is mounted and select Offine.

- 10. Enable the service automatic start feature.
 - Windows

From the Start menu, select Administrative Tools > Services, and select NNM Trap Receiver and NNM Trap Receiver Manager, and then set Startup type to Automatic.

• RHEL 6

ln -s /etc/init.d/nettrap /etc/rc0.d/K01nettrap
ln -s /etc/init.d/nettrap /etc/rc1.d/K01nettrap
ln -s /etc/init.d/nettrap /etc/rc2.d/K01nettrap
ln -s /etc/init.d/nettrap /etc/rc3.d/S98nettrap
ln -s /etc/init.d/nettrap /etc/rc5.d/S98nettrap
ln -s /etc/init.d/nettrap /etc/rc6.d/K01nettrap

• Other Linux distributions

systemctl enable netmgt.service systemctl enable nettrap.service systemctl start netmgt.service systemctl start nettrap.service The NNMi services are started when the systemctl start command is executed.

Note

If you will be running NNMi on the previous active cluster node, the preparations are now complete.

Execute ovstart to start NNMi.

^{19.} Configuring NNMi in a High Availability Cluster

The information provided below applies if either of the following is applicable:

- You will be running NNMi on the previous passive cluster node using a database migrated from an HA configuration to a single-server configuration.
- You used nnmchangeembdbpw.ovpl in the HA configuration to change the database password.
- 11. Use the command shown below to back up the NNMi configuration on the previous active cluster node. This provides a backup containing the data that was copied from the shared disk to a local disk in step 7.
 - Windows

```
%NnmInstallDir%bin\nnmbackup.ovpl -type offline -scope all -target directory
```

• Linux

/opt/OV/bin/nnmbackup.ovpl -type offline -scope all -target directory

- 12. Restore the active cluster node backup data obtained in step 11 to the previous passive cluster node on which you want to run NNMi using the data that was used with the HA configuration.
 - Windows

%NnmInstallDir%bin\nnmrestore.ovpl -force -source backup_data

• Linux

/opt/OV/bin/nnmrestore.ovpl -force -source backup_data

For details about this command, see 20. NNMi Backup and Restore Tools.

^{19.} Configuring NNMi in a High Availability Cluster

19.8.1 Common configuration mistakes

Some common HA configuration mistakes are listed here:

- Disk configuration is not valid.
 - VCS or SCS: If a resource cannot be probed, there is something wrong with the configuration. If a disk cannot be probed, the disk might no longer be accessible by the operating system.
 - Test the disk configuration manually and confirm against HA products documentation that the configuration is appropriate.
- The disk is in use and cannot be started for the HA resource group. Always check that the disk is not activated before starting the HA resource group.
- WSFC network configuration is not valid.

If network traffic is flowing across multiple NIC cards, RDP sessions fail when activating programs that consume a large amount of network bandwidth, such as the NNMi ovjboss process.

- Some HA products do not restart automatically at startup. Review the HA product documentation for details about how to configure automatic restart at startup.
- NFS or other access is added directly to the OS. The resource group configuration must manage this behavior.
- Being in the shared disk mount point during a failover or when the HA resource group is being placed offline. HA kills any processes that prevent the shared disk from being unmounted. Move to a different directory when a failover occurs or when the resource group becomes offline.
- Reusing the HA cluster virtual IP address as the HA resource virtual IP address. This works on one system and not the other. Configure different IP addresses to each system.
- Timeouts are too frequent.

If the products are misbehaving, the HA product might timeout the HA resource and cause a failover. In WSFC, check the value of the **Time to wait for resource to start** setting. NNMi sets this value to 15 minutes, but you can increase it.

• Maintenance mode is not being used.

Maintenance mode was created for debugging HA failures. If you attempt to bring a resource group online on a system and it fails over shortly thereafter, use the maintenance mode to keep the resource group online to see what is failing.

• Cluster logs are not being used.

Cluster logs can show many common mistakes.

19.8.2 HA resource testing

This subsection describes the general approach to testing the resources that you will place into the NNMi HA resource group.

^{19.} Configuring NNMi in a High Availability Cluster

This testing identifies hardware configuration problems. We recommend that you perform this testing before configuring NNMi to run under HA. Keep a record of the configuration values that generate positive results, and use these value when you perform full configuration of the NNMi HA resource group.

For specific details regarding any of the commands listed here, see the most recent documentation for your HA product.

To test HA resources:

- 1. Start the HA cluster.
- 2. (Windows only) Verify that the following virtual IP addresses have been defined for the HA cluster:
 - Virtual IP address for the HA cluster
 - Virtual IP address for each HA resource group

None of these IP addresses can be in use elsewhere.

3. Add an HA resource group to the HA cluster.

Use a non-production name, such as test, for this HA resource group.

- 4. Test the connection to the HA resource group:
 - Add the virtual IP address and corresponding virtual host name for the resource group as a resource to the HA resource group.

Use the values that you will later associate with the NNMi HA resource group.

- Trigger failover from the active cluster node to the passive cluster node to verify that the HA cluster fails over correctly.
- Trigger failover from the new active cluster node to the new passive cluster node to verify failback.
- If the resource group does not fail over correctly, log on to the active node and verify that the IP address is properly configured and accessible. Also verify that no firewall is blocking the IP address.
- Trigger failover from the active cluster node to the passive cluster node to verify that the HA cluster fails over correctly.
- Trigger failover from the new active cluster node to the new passive cluster node to verify failback.
- If the resource group does not fail over correctly, log on to the active cluster node, and verify that the disk is mounted and accessible.
- 5. Keep a record of the commands and inputs that you used to configure the shared disk.

You might need this information when configuring the NNMi HA resource group.

- 6. Remove the resource group from each node.
 - Remove the IP address entry.
 - Offline the resource group, and then remove the resource group from the node

At this point, you can use the NNMi-provided tools to configure NNMi to run under HA.

19.8.3 General HA troubleshooting

(1) Resource hosting subsystem process stops unexpectedly

Starting an HA cluster resource on a computer running the Windows Server 2012, Windows Server 2012 R2 or Windows Server 2016 operating system stops the resource hosting subsystem (rhs.exe) process unexpectedly.

^{19.} Configuring NNMi in a High Availability Cluster

```
http://support.microsoft.com/kb/978527
```

Important

Always run the NNMi resource in a separate resource monitor (rhs.exe) specific to the resource group.

(2) Product monitoring times out

The system log contains a message similar to the following example:

```
VCS ERROR V-16-2-13027 Thread(...) Resource (<resource group>-app) - monitor procedure did not complete within the expected time.
```

This message indicates that the product could not monitor the resources within the time set in Veritas Cluster Server or Symantec Cluster Server.

A timeout value of 60 seconds is set as the default for Veritas Cluster Server or Symantec Cluster Server.

To change the timeout value set in Veritas Cluster Server or Symantec Cluster Server, run the following commands (in the order shown here):

```
/opt/VRTSvcs/bin/haconf -makerw
/opt/VRTSvcs/bin/hares -override resource-group-app MonitorTimeout
/opt/VRTSvcs/bin/hares -modify resource-group-app MonitorTimeout <value in seconds>
/opt/VRTSvcs/bin/haconf -dump -makero
```

(3) 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.

(4) Cannot start the NNMi HA resource group on a particular cluster node

If the nnmhargconfigure.ovpl or nnmhastartrg.ovpl command does not correctly start, stop, or switch the NNMi HA resource group, review the following information:

- WSFC
 - Review in Failover Cluster Management the state of the resource group and underlying resources.

- Review the Event Viewer log for any errors.

- VCS or SCS
 - 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 start the NNMi HA resource group manually by using HA product commands:

1. Mount the shared disk.

^{19.} Configuring NNMi in a High Availability Cluster

- 2. Assign the virtual host to the network interface:
 - WSFC
 - Start Failover Cluster Management.
 - Expand the resource group.
 - Right-click *resource-group*-ip, and then click **Bring Online**.
 - VCS or SCS

/opt/VRTSvcs/bin/hares -online resource-group-ip -sys local-host-name

3. Start the HA resource group.

Example:

• Windows

%NnmInstallDir%misc\nnm\ha\nnmhastartrg.ovpl NNM -start resource-group

• Linux

\$NnmInstallDir/misc/nnm/ha/nnmhastartrg.ovpl NNM -start resource-group

Return code 0 indicates that NNMi started successfully.

Return code 1 indicates that NNMi did not start correctly.

(5) The message System error XXXX occurred is displayed (in Windows)

A system (OS or cluster software) error might have occurred. For details, see the OS or cluster software documentation.

Error examples: Examples of errors in WSFC are described below.

- Example: System error 5054 occurred (0x000013be). The cluster network is invalid. If an IP address of the internal network for heartbeat is specified as an IP address for NNMi, this error occurs in the cluster.exe command that was executed to create IP address resources.
- Example: System error 5057 occurred (0x000013c1). That cluster IP address is already in use.

If an IP address already in use is specified as an IP address for NNMi, this error occurs in the cluster.exe command that was executed to create IP address resources.

Action: Check the nature of the system error and take appropriate action. When a specified IP address is not valid for NNMi, as is the case in these examples, check the IP address that is to be used.

19.8.4 NNMi-specific HA troubleshooting

The topics in this subsection apply to HA configuration for NNMi only.

(1) NNMi does not start correctly under HA

When NNMi does not start correctly, you must determine whether the issue is a hardware issue with the virtual IP address or the disk, or whether the issue is some form of application failure. During this determination process, put the system in maintenance mode.

To fix this problem:

^{19.} Configuring NNMi in a High Availability Cluster

1. On the active cluster node in the HA cluster, disable HA resource group monitoring by creating the following maintenance file:

Windows: %NnmDataDir%hacluster*resource-group*\maintenance

Linux: \$NnmDataDir/hacluster/resource-group/maintenance

2. Start NNMi:

ovstart

3. Verify that NNMi started correctly:

ovstatus -c

All NNMi services must show the state RUNNING. If this is not the case, troubleshoot the process that does not start correctly.

4. After completing your troubleshooting, delete the maintenance file:

Windows: %NnmDataDir%hacluster*resource-group*\maintenance

Linux: \$NnmDataDir/hacluster/resource-group/maintenance

(2) Changes to NNMi data are not seen after failover

The NNMi configuration points to a different system than the one NNMi is running. To fix this problem, verify that the ov.conf file has appropriate entries for the following items:

- NNM_INTERFACE=virtual-host-name
- HA_RESOURCE_GROUP=resource-group
- HA_MOUNT_POINT=HA-mount-point
- NNM_HA_CONFIGURED=YES
- HA_POSTGRES_DIR=*HA-mount-point*/NNM/dataDir/shared/nnm/databases/Postgres
- HA_CUSTOMPOLLER_DIR=*HA-mount-point*/NNM/dataDir/shared/nnm/databases/ custompoller
- HA_NNM_LOG_DIR=*HA-mount-point*/NNM/dataDir/log/nnm
- HA_JBOSS_DATA_DIR=*HA-mount-point*/NNM/dataDir/nmsas/NNM/data
- HA_LOCALE=locale (Linux only)
- HA_PERFSPI_ADAPTER_DIR=HA-mount-point/NNM/dataDir/shared/perfSpi/datafiles

For the location of the ov.conf file, see 19.9.1 NNMi HA configuration files.

(3) nmsdbmgr does not start after HA configuration

This situation usually occurs as a result of starting NNMi after running the nnmhaconfigure.ovpl command but without having run the nnmhadisk.ovpl command with the -to option specified. In this case, the HA_POSTGRES_DIR entry in the ov.conf file specifies the location on the shared disk, but this location is not available to NNMi.

To fix this problem:

1. On the active cluster node in the HA cluster, disable HA resource group monitoring by creating the following maintenance file:

^{19.} Configuring NNMi in a High Availability Cluster

- Windows: %NnmDataDir%hacluster\resource-group\maintenance
- Linux: \$NnmDataDir/hacluster/*resource-group*/maintenance
- 2. Copy the NNMi database to the shared disk:
 - Windows: %NnmInstallDir%misc\nnm\ha\nnmhadisk.ovpl NNM -to HA-mount-point
 - Linux: \$NnmInstallDir/misc/nnm/ha/nnmhadisk.ovpl NNM -to HA-mount-point
- 3. Start the NNMi HA resource group:
 - Windows: %NnmInstallDir%misc\nnm\ha\nnmhastartrg.ovpl NNM resource-group
 - Linux: \$NnmInstallDir/misc/nnm/ha/nnmhastartrg.ovpl NNM resource-group

4. Start NNMi:

ovstart

5. Verify that NNMi started correctly:

```
ovstatus -c
```

All NNMi services must show the state RUNNING.

6. After completing your troubleshooting, delete the maintenance file:

- Windows: %NnmDataDir%hacluster*resource-group*\maintenance
- Linux: \$NnmDataDir/hacluster/resource-group/maintenance

(4) NNMi runs correctly on only one HA cluster node (Windows)

The Windows operating system requires two different virtual IP addresses, one for the HA cluster and one for the HA resource group. If the virtual IP address of the HA cluster is the same as that of the NNMi HA resource group, NNMi runs correctly only on the node associated with the HA cluster IP address.

To correct this problem, change the virtual IP address of the HA cluster to a unique value within the network.

(5) Disk failover does not occur

This situation can arise when the operating system does not support the shared disk. Review the HA product, operating system, and disk manufacturer documentation to determine whether these products can work together.

When a disk failure occurs, NNMi does not start on failover. Most likely, nmsdbmgr fails because the HA_POSTGRES_DIR directory does not exist. Verify that the shared disk is mounted and that the appropriate files are accessible.

(6) Shared disk is not accessible (Windows)

If nothing is displayed even after the command nnmhaclusterinfo.ovpl -config NNM -get HA_MOUNT_POINT is run, this indicates that the shared disk cannot be accessed because the specified mount point is incorrect.

The drive of the shared disk mount point must be fully specified during HA configuration.

Example: Y:

To correct this problem, run the nnmhaconfigure.ovpl command on each node in the HA cluster. Fully specify the drive of the shared disk mount point.

^{19.} Configuring NNMi in a High Availability Cluster

(7) Shared disk files are not found on the secondary cluster node after failover

The most common cause of this situation is that the nnmhadisk.ovpl command was run with the -to option specified while 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.

To fix this problem:

1. On the active cluster node in the HA cluster, disable HA resource group monitoring by creating the following maintenance file:

Windows: %NnmDataDir%hacluster\resource-group\maintenance

Linux: \$NnmDataDir/hacluster/resource-group/maintenance

- 2. Log on to the active cluster node and verify that the disk is mounted and available.
- 3. Stop NNMi:

ovstop

Windows: net stop NnmTrapReceiver

Linux:

- For distributions that uses systemd to manage services

/opt/OV/bin/nettrap stop

- Other distributions

/etc/init.d/nettrap stop

4. Copy the NNMi database to the shared disk:

Windows: %NnmInstallDir%misc\nnm\ha\nnmhadisk.ovpl NNM -to *HA-mount-point* Linux: \$NnmInstallDir/misc/nnm/ha/nnmhadisk.ovpl NNM -to *HA-mount-point*

5. Start the NNMi HA resource group:

Windows: %NnmInstallDir%misc\nnm\ha\nnmhastartrg.ovpl NNM resource-group Linux: \$NnmInstallDir/misc/nnm/ha/nnmhastartrg.ovpl NNM resource-group

6. Start NNMi:

ovstart

7. Verify that NNMi started correctly:

ovstatus -c

All NNMi services must show the state RUNNING.

8. After completing your troubleshooting, delete the maintenance file: Windows: %NnmDataDir%hacluster\resource-group\maintenance

windows. SMinipatabilishactuster (resource-group (maintenance

Linux: \$NnmDataDir/hacluster/resource-group/maintenance

^{19.} Configuring NNMi in a High Availability Cluster

This section explains the NNMi HA configuration files, scripts, and log files.

19.9.1 NNMi HA configuration files

The table below lists the NNMi HA configuration files. These files apply to NNMi and are installed at the following location:

• Windows

%NnmDataDir%shared\nnm\conf

• Linux

\$NnmDataDir/shared/nnm/conf

Table 19-11: NNMi HA configuration files

File name	Description
ov.conf	This file describes the NNMi HA implementation and is updated by the nnmhaclusterinfo.ovpl command. NNMi processes read this file to determine the HA configuration.
nnmdatareplicator.conf	This file is used by the nnmdatareplicator.ovpl command to determine the NNMi folders and files that have been included in data replication from the active cluster node to the passive cluster nodes. If you implement a different method of replicating the NNMi configuration, see this file for a list of the data to include. For details, see the comments in this file.

19.9.2 NNMi-provided HA configuration scripts

The tables below list 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 commands provided by the HA product to configure HA for NNMi.

On the NNMi management server, the NNMi-provided HA configuration scripts are installed at the following location:

- Windows %NnmInstallDir%misc\nnm\ha
- Linux

\$NnmInstallDir/misc/nnm/ha

Table 19-12:	NNMi HA	configuration	scripts
		<u> </u>	

Script name	Description
nnmhaconfigure.ovpl	Configures NNMi for an HA cluster. Run this script on all nodes in the HA cluster.
nnmhaunconfigure.ovpl	Unconfigures NNMi from an HA cluster. Run this script on one or more nodes in the HA cluster, as needed.
nnmhaclusterinfo.ovpl	Retrieves cluster information regarding NNMi. Run this script as needed on any node in the HA cluster.

^{19.} Configuring NNMi in a High Availability Cluster

Script name	Description
nnmhadisk.ovpl	Copies NNMi data files to and from the shared disk. During HA configuration, run this script on the primary cluster node. At other times, run this script per the instructions in this chapter.
nnmhastartrg.ovpl	Starts the NNMi HA resource group in an HA cluster. During HA configuration, run this script on the primary cluster node.
nnmhastoprg.ovpl	Stops the NNMi HA resource group in an HA cluster. During HA unconfiguration, run this script on the active cluster node.

The NNMi-provided scripts listed in Table 19-13 are used by the scripts listed in Table 19-12. Do not run the scripts listed in Table 19-13 directly.

Table 19-13: NNMi HA support scripts

Script name	Description
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. For VCS or SCS configurations, this script is used by the VCS or SCS start, stop, and monitor scripts (nnmhargconfigure.ovpl configures this usage). This script is also used by nnmhastartrg.ovpl to enable and disable tracing.
nnmhargconfigure.ovpl	Configures HA resources and resource groups. This script is used by nnmhaconfigure.ovpl and nnmhaunconfigure.ovpl.
nnmhastart.ovpl	Starts NNMi in an HA cluster. This script is used by nnmharg.ovpl.
nnmhastop.ovpl	Stops NNMi in an HA cluster. This script is used by nnmharg.ovpl.
nnmhamonitor.ovpl	Monitors NNMi processes in an HA cluster. This script is used by nnmharg.ovpl.
nnmhamscs.vbs	This is a template for creating a script to start, stop, and monitor NNMi processes in a WSFC HA cluster. The generated script is used by WSFC and is stored at the following location: %NnmDataDir%hacluster\resource-group\hamscs.vbs

19.9.3 NNMi HA configuration log files

The following log files are applicable to an HA configuration for NNMi:

- Windows configuration
 - NnmDataDir%tmp\HA_nnmhaserver.log
 - -NnmDataDir%log\haconfigure.log
- Linux
 - \$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\postgres.log

^{19.} Configuring NNMi in a High Availability Cluster

- %HA MOUNT POINT%\NNM\dataDir\log\nnm\nmsdbmgr.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, and starting and stopping issues.

• Linux for VCS or SCS

Resource	Log file
<i>resource-group</i> -app	 /var/VRTSvcs/log/Application_A.log \$HA_MOUNT_POINT/NNM/dataDir/log/nnm/ovspmd.log \$HA_MOUNT_POINT/NNM/dataDir/log/nnm/public/postgres.log \$HA_MOUNT_POINT/NNM/dataDir/log/nnm/public/nmsdbmgr.log Linux: /var/log/messages* Solaris: /var/adm/messages*
<pre>resource-group-dg resource-group-volume resource-group-mount</pre>	 /var/VRTSvcs/log/DiskGroup_A.log /var/VRTSvcs/log/Volume_A.log /var/VRTSvcs/log/Mount_A.log Linux: /var/log/messages* Solaris: /var/adm/messages*
<i>resource-group</i> -ip	 /var/VRTSvcs/log/IP_A.log Linux: /var/log/messages* Solaris: /var/adm/messages*

Note: For operating system-specific issues related to the HA resources, review the /var/adm/messages* or / var/log/messages* files. For *resource-group*-app, look for messages regarding unable to start a process.

^{19.} Configuring NNMi in a High Availability Cluster

Part 6: NNMi Maintenance



NNMi Backup and Restore Tools

Good backup and restore strategies are key to ensuring the uninterrupted functioning of any business. Because NNMi is an important asset for network operations, it must be backed up regularly. The two types of critical data related to an NNMi installation are as files in the file system and data in the relational database. This chapter explains the tools NNMi provides for backing up and restoring important NNMi files and data.

20.1 Backup and restore commands

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 NNMi database.

• nnmrestore.ovpl

Restores a backup that was created with the nnmbackup.ovpl script.

• nnmbackupembdb.ovpl

Creates a complete backup of the NNMi database (but not the file system data) while NNMi is running.

- nnmrestoreembdb.ovpl Restores a backup that was created with the nnmbackupembdb.ovpl script.
- nnmresetembdb.ovpl Drops the NNMi database tables. Run the ovstart command to re-create the tables.

For each command's syntax, see the appropriate reference page.

^{20.} NNMi Backup and Restore Tools

20.2 Backing up NNMi data

The NNMi backup command (nnmbackup.ovpl) copies key NNMi file system data and some or all of the tables in the NNMi Postgres database to a specified target directory. Each backup operation stores files in a parent directory called nnm-bak-*timestamp* inside the target directory. You can specify the -noTimeStamp option to save disk space, in which case the parent directory is named simply nnm-bak. When a backup is performed after a previous backup with the -noTimeStamp option in effect, the previous backup is renamed nnm-bak.previous, thereby creating a rolling backup. This renaming occurs after the second backup has been completed to protect against any loss of backup data.

The NNMi backup command can create a tar archive of the backup data. You can use any tool of your choice to compress backup files. You can also use any appropriate tool to save a copy of a backup. For details, see the *nnmbackup.ovpl Reference Page*.

20.2.1 Backup type

The NNMi backup command supports two types of backups:

- Online backups occur while NNMi is running. NNMi ensures that the database tables are synchronized in the backed up data. Operators can be actively using the NNMi console and other processes can be interacting with the NNMi database during an online backup. With an online backup, you can back up all NNMi data or only some of the data according to function, as described in 20.2.2 Backup scope. For the NNMi database, the nmsdbmgr service must be running.
- Offline backups occur while NNMi is completely stopped. With an offline backup, the backup scope applies to the file system files only. An offline backup always includes the complete NNMi database regardless of the backup scope. For the NNMi database, the backup copies the Postgres database files.

20.2.2 Backup scope

The NNMi backup command provides several scopes that define how much of NNMi is backed up.

Configuration scope

The configuration scope (-scope config) loosely aligns to the information in the **Configuration** workspace on the NNMi console.

The configuration scope includes the following data:

- For online backups, only those database tables that store NNMi configuration information.
- For offline backups, the entire database.
- For all backups, the NNMi configuration information in the file system, as listed in Table 20-1: Configuration scope files and directories.

Topology scope

The topology scope (-scope topology) loosely aligns to the information in the **Inventory** workspace on 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.

The topology scope includes the following data:

- For online backups, only those database tables that store NNMi configuration and network topology information.
- For offline backups, the entire database.

• For all backups, the NNMi configuration information in the file system, as listed in Table 20-1: Configuration scope files and directories. Currently, there are no file system files associated with the topology scope.

Event scope

The event scope (-scope event) loosely aligns to the information in the **Incident Browsing** workspace on the NNMi console. Because events are dependent on the network topology related to those events, the event scope includes the configuration and topology scopes.

The event scope includes the following data:

- For online backups, only those database tables that store NNMi configuration, network topology, and event information.
- For offline backups, the entire database.
- For all backups, the NNMi configuration information in the file system, as listed in Table 20-1: Configuration scope files and directories, and the NNMi event information, as listed in Table 20-2: Event scope files and directories.

All scope

The complete backup (-scope all) includes all important NNMi files and the complete database.

Table 20-1: Configuration scope files and directories

Directory or file name	Description
%NnmInstallDir%conf (Windows only)	Configuration information
%NnmInstallDir%misc\nms\lic \$NnmInstallDir/misc/nms/lic	Miscellaneous license information
%NnmDataDir%nmsas\NNM\conf \$NnmDataDir/nmsas/NNM/conf	JBoss configuration
%NnmDataDir%conf \$NnmDataDir/conf	Configuration information
%NnmDataDir%conf\nnm\props \$NnmDataDir/conf/nnm/props	Local NNMi configuration properties files
%NnmDataDir%shared\nnm\conf\licensing\LicFile.txt \$NnmDataDir/shared/nnm/conf/licensing/LicFile.txt	License information
%NnmDataDir%NNMVersionInfo \$NnmDataDir/NNMVersionInfo	NNMi version information file
%NnmDataDir%shared\nnm\user-snmp-mibs \$NnmDataDir/shared/nnm/user-snmp-mibs	Shared user-added SNMP MIB information
%NnmDataDir%shared\nnm\actions \$NnmDataDir/shared/nnm/actions	Shared lifecycle transition actions
%NnmDataDir%shared\nnm\certificates \$NnmDataDir/shared/nnm/certificates	Shared NNMi SSL certificates
%NnmDataDir%shared\nnm\conf \$NnmDataDir/shared/nnm/conf	Shared NNMi configuration information
%NnmDataDir%shared\nnm\conf\licensing \$NnmDataDir/shared/nnm/conf/licensing	Shared NNMi license configuration information
%NnmDataDir%shared\nnm\lrf \$NnmDataDir/shared/nnm/lrf	Shared NNMi component registration files

Directory or file name	Description
%NnmDataDir%shared\nnm\conf\props \$NnmDataDir/shared/nnm/conf/props	Shared NNMi configuration properties files
%NnmDataDir%shared\nnm\www\htdocs\images \$NnmDataDir/shared/nnm/www/htdocs/images	Shared background images for NNMi node group map

In this context, files in the shared directories are those shared with another NNMi management server in an NNMi application failover or high availability environment.

Table 20-2: Event scope files and directories

Directory or file name	Description
%NnmDataDir%log\nnm\signin.log	NNMi console sign-in log
<pre>\$NnmDataDir/log/nnm/signin.log</pre>	

20. NNMi Backup and Restore Tools

20.3 Restoring NNMi data

The NNMi restore script (nnmrestore.ovpl) places backup data on the NNMi management server. The type and scope of the backup determines what NNMi can restore.



If you use the nnmrestore.ovpl script to place database records on a second NNMi management server, both NNMi management servers must have the same type of operating system and the same NNMi version and patch level.

Important

Do not restore backup data acquired from an NNMi with a cluster configuration to an NNMi with a single configuration.

When you are using the global network management function, placing the backup data from one NNMi management server onto a second NNMi management server means that both servers have the same database UUID. After you restore NNMi on the second NNMi management server, you must uninstall NNMi from the original NNMi management server.

- In restoring an online backup, NNMi copies the file system data to the correct locations and overwrites the contents of the database tables that were included in the backup. Objects that were deleted after the backup was made are restored, and objects that have been created since the backup was made disappear. Additionally, any objects that were changed after the backup was made revert to their state at the time of the backup. For the NNMi database, the nmsdbmgr service must be running.
- In restoring an offline backup, NNMi overwrites the Postgres files in the file system, completely replacing the database files with the contents of the backup.

With the -force option, the nnmrestore.ovpl command stops all NNMi processes, starts the nmsdbmgr service (if restoring from an online backup of the NNMi database), restores the data, and then restarts all NNMi processes.

If the provided source is a tar file, the NNMi 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 capacity to support the temporary folder, or extract the archive before running the restore command.

🖹 Note

Because the database schema might change from one version of NNMi to the next, data backups cannot be shared across versions of NNMi.

20.3.1 Same-system restore

When you use the backup and restore commands on the same system (single-system backup and restore), the following items must not have changed between the time of the backup and the time of the restore:

- NNMi version (including any patches)
- Operating system type

20. NNMi Backup and Restore Tools

- Character set (language)
- Host name
- Domain

20.3.2 Different-system restore

You can also use the backup and restore commands to transfer data from one NNMi management server to another. The intended uses of different-system restoration include recovering from a system failure and transferring NNMi to a different system during an operating system upgrade.



You must note the following if you are using the global network management function: Because the NNMi UUID is copied to the target system during the database restore, both the source system and the target system will appear to be running the same instance of NNMi. To resolve this anomaly, you must uninstall NNMi from the source system.

Note

To create multiple functional NNMi management servers with similar configurations, such as while deploying global network management, use the nnmconfigexport.ovpl and nnmconfigimport.ovpl commands.

For a different-system restore, the following items must be identical on both systems:

- NNMi version (including any patches)
- Operating system type
- Character set (language)

The following items can differ between the two systems:

- Host name
- Domain

In a different-system restore, the nnmrestore.ovpl command does not copy license information to the new system. You must obtain and apply a new license for the new NNMi management server. For details, see the license manual.

20.4.1 Back up all data periodically

Make sure that your disaster recovery plan includes a regularly scheduled complete backup of all NNMi data. You do not need to shut down NNMi to create this backup. If you incorporate the backup into a script, you can use the -force option to ensure that NNMi is in the correct state before the backup begins.

Example:

nnmbackup.ovpl -force -type online -scope all -archive -target nnmi_backups\periodic

To recover your NNMi data after a hardware failure, follow these steps:

- 1. Repair or replace the failed hardware.
- 2. Install NNMi to the same version and patch level as were in place at the time of the backup.
- 3. Restore the NNMi data.
 - If the recovery NNMi management server satisfies the requirements listed in 20.3.1 Same-system restore, run a command similar to the following example:

nnmrestore.ovpl -force -lic -source nnmi backups\periodic\newest backup

• If the recovery NNMi management server does not qualify for a same-system restore but satisfies the requirements listed in 20.3.2 Different-system restore, run a command similar to the following example:

nnmrestore.ovpl -force -source nnmi backups/periodic/newest backup

Update the licensing as needed.

20.4.2 Back up data before changing the configuration

Perform a scoped backup (as described in 20.2.2 Backup scope) 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 operational configuration.

Example:

nnmbackup.ovpl -type online -scope config -target nnmi_backups\config

To restore this backup to the same NNMi management server, stop all NNMi processes, and then run a command similar to the following example:

nnmrestore.ovpl -force -source nnmi_backups\config\newest_backup

```
20. NNMi Backup and Restore Tools
```
20.4.3 Back up data before upgrading NNMi or the operating system

Before making major system changes (including upgrading NNMi or the operating system), perform a complete backup of all NNMi data. To ensure that no changes are made to the NNMi database after the backup is made, stop all NNMi processes and create an offline backup.

Example:

nnmbackup.ovpl -type offline -scope all -target nnmi backups\offline

If NNMi does not run correctly after the system change, roll back the change or set up a different NNMi management server and ensure that the requirements listed in 20.3.2 Different-system restore are satisfied, and then run a command similar to the following example:

Example:

```
nnmrestore.ovpl -source nnmi backups\offline\newest backup
```

Update the licensing as needed.

20.4.4 Restore file system files only

To overwrite NNMi files without affecting the database tables, run a command similar to the following example:

Example:

nnmrestore.ovpl -partial -source nnmi_backups\offline\newest_backup

^{20.} NNMi Backup and Restore Tools

20.5 Backing up and restoring the database

NNMi provides the nnmbackupembdb.ovpl and nnmrestoreembdb.ovpl commands to back up and restore the NNMi database only. This functionality is useful for creating a snapshot of the data as you experiment with NNMi configuration settings.

The nnmbackupembdb.ovpl command performs online backups only. At a minimum, the nmsdbmgr service must be running.

Each backup operation stores files in a parent directory called nnm-bak-<TIMESTAMP> in the target directory. You can specify the -noTimeStamp option to save disk space. When you use the -noTimeStamp option, the parent directory is simply named nnm-bak. When a backup is performed after a previous backup was made using the -noTimeStamp option, the previous backup is renamed nnm-bak.previous, thereby creating rolling backups. This renaming is performed after the second backup has been made to protect against any loss of backup data.

Q Тір

Run the nnmresetembdb.ovpl command before restoring data to the database. This command ensures that the database does not contain any errors, thus eliminating the possibility of encountering database constraint violations. For details about running this database reset command, see the *nnmresetembdb.ovpl Reference Page*.



Maintaining NNMi

Once the NNMi management servers begin working, you can perform maintenance tasks in order to optimize several NNMi features.

21.1 Administering access control lists for NNMi folders

You might encounter a situation that would cause you to modify the user name that runs the NNM Action Server. However, if you change the user name that runs the action server without modifying the user name permissions, the NNM Action Server might not start, and NNMi might not log messages when running incident actions. This section discusses the actions to take to prevent this from happening.

NNMi supports changing the permissions for the following directories:

- /var/opt/OV/log/nnm/public
- /var/opt/OV/shared/perfSpi

Although the default permissions for the /var/opt/OV/log/nnm/public folder are 755, NNMi uses ACLs to adjust access permissions for the database user (nmsdbmgr) and the nnmaction user (bin). During NNMi post-installation (part of the installation or upgrade script), the installation script changes the /var/opt/OV/log/nnm/ public folder permissions and adds the ACLs.

If the installation script is unable to set the ACLs in the /var/opt/OV/log/nnm/public folder due to some unexpected error, the script will leave the /var/opt/OV/log/nnm/public folder world-writable (by other users), even though the NNMi installation completes successfully. Following a successful NNMi installation, if you want to restrict world-write permissions on the /var/opt/OV/log/nnm/public folder, see the system administrator's documentation to determine how to set up ACLs for the NNMi management server's operating system.

For the /var/opt/OV/log/nnm/public folder, use Linux ACLs (access control lists) to adjust user access. Configuring ACLs is a useful method for extending the owner/group/other permissions. ACLs are supported in Linux.

For example, after running the command listed below, the user depicted by the *user* variable obtains write access to the folder /var/opt/OV/log/nnm/public. Without running the following command, the permissions for the /var/opt/OV/log/nnm/public folder are 755, and files within the directory are not writable by anyone other than root.

setfacl -m user:user:rwx /var/opt/OV/log/nnm/public

For details about how to use the setfacl command, see the appropriate reference pages.

JP1/Network Node Manager i Setup Guide

NNMi provides a command line tool to help you automate the configuration of node groups. The nnmnodegroup.ovpl command enables you to create, list, modify, and delete node groups.

For details, see the *nnmnodegroup.ovpl Reference Page*.

^{21.} Maintaining NNMi

21.3 Configuring node group map settings

In addition to using the NNMi console to configure node group map settings, you can use the nnmnodegroupmapsettings.ovpl command line tool to configure node group map settings. This tool lets you create, modify, and delete node group map settings. It also lets you list current node group map settings in TXT, XML, or CSV format.

Note

You can refresh the Web browser in which NNMi is currently running to see immediately the effects of the changes made to your node group map settings.

For details, see the nnmnodegroupmapsettings.ovpl Reference Page.

21.4 Configuring communication settings

You can use the nnmcommunication.ovpl command line tool to configure NNMi communication settings. This tool lets you create, list, modify, and delete communication settings. It can generate lists in text tables, as text lists, or in XML format.

An administrator can also use the nnmcommunication.ovpl tool to lock and directly manage SNMP agent settings for such fields as management addresses and community strings, bypassing the normal configuration.

The nnmcommunication.ovpl tool supports creating, updating, or deleting SNMP proxy ports and SNMP proxy addresses for Default, Node-specific, Region-specific, and SNMP Agent-specific settings using the Command Line Interface (CLI).

For details, see the *nnmcommunication.ovpl Reference Page*.

21.5 Administering a Custom Poller collection export

The Custom Poller feature enables you to take a proactive approach to network management by using SNMP MIB expressions to specify additional information that NNMi needs to poll. A Custom Poller collection defines the information you want to gather (poll) as well as how NNMi is to react to the gathered data. For details, see *Create a Custom Poller Collection* and Create Custom Polling Configurations in NNMi Help.

The Custom Poller feature relies on you to remove files from the export directory as you process them. Do not use the exported files for long-term storage; if they consume more than the configured maximum disk space, NNMi will remove the older files as it creates new ones. Unless you process these files and store them in a different location, you will lose them.

21.5.1 Changing the Custom Poller collections export directory

NNMi writes the data from the collections you export into the following directory:

- Windows: %NNM DATA%\shared\nnm\databases\custompoller\export
- Linux: \$NNM_DATA/shared/nnm/databases/custompoller/export

To change the directory into which NNMi writes its Custom Poller files, follow these steps:

1. Edit the following file:

- Windows: %NNM_PROPS%\nms-custompoller.properties
- Linux: \$NNM_PROPS/nms-custompoller.properties
- 2. Look for the exportdir entry.

This entry is similar to the following line:

#!com.hp.nnm.custompoller.exportdir=base-directory-to-export-custom-poller-metrics

To configure NNMi to write Custom Poller collection information into the C: \CustomPoller directory, change the line as follows:

com.hp.nnm.custompoller.exportdir=C:/CustomPoller

Make sure that the characters #! at the beginning of the line are deleted.

Important

For the directory delimiter character, use a forward slash (/), not a backslash (\setminus), in Windows as well.

- 3. Save your changes.
- 4. Restart NNMi by running the following commands:

```
ovstop
ovstart
```

21.5.2 Changing the maximum amount of disk space for Custom Poller collections export

To change the maximum amount of disk space that NNMi uses when exporting data to collection_name.csv files, follow these steps:

- 1. Edit the following file:
 - Windows: %NNM_PROPS%\nms-custompoller.properties
 - Linux: \$NNM_PROPS/nms-custompoller.properties
- 2. Look for the maxdiskspace entry.

This entry is similar to the following line:

```
#!com.hp.nnm.custompoller.maxdiskspace=1000
```

To configure NNMi to reserve up to 2,000 megabytes (2 gigabytes) of storage space for each collection name.csv file, change the line as follows:

```
com.hp.nnm.custompoller.maxdiskspace=2000
```

- 3. Save your changes.
- 4. Restart NNMi by running the following commands:

```
ovstop
ovstart
```

21.5.3 Changing the Custom Poller metric accumulation interval

NNMi sets the period of time (in minutes) during which it accumulates Custom Poller Collection metrics before it writes data into a file. To change the Custom Poller metric accumulation interval, follow these steps:

1. Edit the following file:

- Windows: %NNM PROPS%\nms-custompoller.properties
- Linux: \$NNM PROPS/nms-custompoller.properties
- 2. Look for a line that resembles the following:

#!com.hp.nnm.custompoller.accumulationinterval=5

To configure NNMi to collect metrics for 10 minutes instead of the default value of 5 minutes, change the line as follows:

com.hp.nnm.custompoller.accumulationinterval=10

- 3. Save your changes.
- 4. Restart NNMi by running the following commands:

```
ovstop
ovstart
```

21.6 Administering incident actions

You can configure actions to run automatically at any point in the incident lifecycle. For example, you might want to configure an action to occur when an incident of the type you are configuring is generated. For details, see *Configure an Action for an Incident* in NNMi Help.

To adjust action parameters, follow the steps shown in the following subsections.

21.6.1 Setting the number of simultaneous actions

To modify the number of simultaneous actions that NNMi can run, follow these steps:

- 1. Edit the following file:
 - Windows: %NNM PROPS%\nnmaction.properties
 - Linux: \$NNM_PROPS/nnmaction.properties
- 2. Look for a line that resembles the following:

#!com.hp.ov.nms.events.action.numProcess=10

To configure NNMi to enable 20 simultaneous actions instead of the default value, change the line as follows:

com.hp.ov.nms.events.action.numProcess=20

Make sure you remove the #! characters at the beginning of the line.

- 3. Save your changes.
- 4. Restart NNMi by running the following commands:

```
ovstop
ovstart
```

21.6.2 Setting the number of threads for Jython actions

To modify the number of threads the action server uses to run jython scripts, follow these steps:

- 1. Edit the following file:
 - Windows: %NNM_PROPS%\nnmaction.properties
 - Linux: \$NNM_PROPS/nnmaction.properties
- 2. Look for a line that resembles the following:

#!com.hp.ov.nms.events.action.numJythonThreads=10

To configure NNMi to enable 20 threads for running jython scripts instead of the default value, change the line as follows:

com.hp.ov.nms.events.action.numJythonThreads=20

Make sure you remove the #! characters at the beginning of the line.

3. Save your changes.

```
21. Maintaining NNMi
```

4. Restart NNMi by running the following commands:

```
ovstop
ovstart
```

21.6.3 Setting the action server name parameter

To modify the user name that runs the action server on a Windows NNMi management server, change the LogOn property of the NNM Action Server service. Specify a user name that has administrator permissions.

To modify the user name that runs the action server from an NNMi management server running on Linux operating system, complete the following steps:

1. Edit the following file:

\$NNM_PROPS/nnmaction.properties

2. Look for a line that resembles the following:

#!com.hp.ov.nms.events.action.userName=bin

To configure NNMi to have system run the action server instead of the default value, change the line as follows:

com.hp.ov.nms.events.action.userName=system

Make sure you remove the #! characters at the beginning of the line.

- 3. Save your changes.
- 4. Restart the action server by running the following commands:
 - a. ovstop nnmaction
 - b. ovstart nnmaction

21.6.4 Changing the action server queue size

If a command that does not terminate for a long time is defined as an incident action for a large number of incidents that occur over a short period of time, the action server might use up a lot of memory. To provide better action server performance, Hitachi places limits on the memory size that the action server can use.

To modify these limits, follow these steps:

- 1. Edit the following file:
 - Windows: %NNM PROPS%\nnmaction.properties
 - Linux: \$NNM_PROPS/nnmaction.properties
- 2. Look for two lines that resemble the following:

```
com.hp.ov.nms.events.action.jvmargs.minMemsize=-Xms6m
com.hp.ov.nms.events.action.jvmargs.maxMemsize=-Xmx30m
```

- 3. These parameters show that the minimum and maximum memory size are set to 6 megabytes and 30 megabytes, respectively. Adjust these parameters to meet your needs.
- 4. Save your changes.

```
21. Maintaining NNMi
```

5. Restart NNMi by running the following commands:

```
ovstop
ovstart
```

21.6.5 Incident actions log

When an action runs, output is logged to the associated incident actions log file. To view the contents of the log for a selected incident, choose **Tools** and then the **Incident Actions Log** menu option. The following table describes the items contained in the log:

Item	Description
Command	Command to run when the lifecycle state in which the incident is configured is reached
Incident Name	Name of incident
Incident UUID	UUID of the incident (displayed under the Registration tab)
Command Type	Type of command (Jython or ScriptOrExecutable)
Lifecycle	Lifecycle state of the incident (Registered, In Process, Completed, or Closed)
Exit Code	Command's return code
Standard Output	Standard output of the action
Standard Error	Standard error output
Execution Status	Determined status of the action

21.7 Overriding settings in the server.properties file

A system might have two server.properties files.

The following file, which is created by the product installer, contains properties that customize the application server for application instances. This file is not modifiable by users and is replaced during code maintenance (upgrades and patches).

- Windows: %NnmInstallDir%NNM\server\server.properties
- Linux: \$NnmInstallDir/NNM/server/server.properties

The following file is used by users to configure applications for their environment and will not be modified by the product during upgrades or patches. This file overrides values configured in other files. All customizing is implemented in this file.

- Windows: %NnmDataDir%nmsas\NNM\server.properties
- Linux: \$NnmDataDir/nmsas/NNM/server.properties

21.7.1 Override the browser locale setting

You can use the following server.properties file to force a specified locale value for all NNMi clients regardless of their browsers' locale value:

- Windows: %NnmDataDir%nmsas\NNM\server.properties
- Linux: \$NnmDataDir/nmsas/NNM/server.properties

When this value is set in the server.properties file, a browser's locale value is ignored.

To override browser locale settings:

1. Open the server.properties file.

- Windows: %NnmDataDir%nmsas\NNM\server.properties
- Linux: \$NnmDataDir/nmsas/NNM/server.properties
- 2. Navigate to nmsas.server.forceClientLocale.
- 3. Set nmsas.server.forceClientLocale to either of the following:
 - nmsas.server.forceClientLocale = two-letter-ISO-language-code

For example, enter the following to use only the ISO language code to set the locale to English:

Example:

nmsas.server.forceClientLocale = en

• nmsas.server.forceClientLocale = two-letter-ISO-language-code two-letter-ISO-country-code

For example, enter the following to use the ISO language and country codes to set the locale to English:

Example:

```
nmsas.server.forceClientLocale = en_US
```

4. Restart the NNMi ovjboss service by running the following commands on the NNMi management server:

```
ovstop ovjboss
ovstart
```

Changes to the server.properties file are read only at the time of ovjboss startup.

For details, see the comments in the server.properties file.

21.7.2 Configuring SNMP Set object access privileges

You can use the following file to configure the object access privileges required for using the SNMP Set feature on the nodes to which users have access:

- Windows: %NnmDataDir%nmsas\NNM\server.properties
- Linux: \$NnmDataDir/nmsas/NNM/server.properties

For details about the SNMP Set feature, see NNMi Help. For details about object access privileges, see NNMi Help for Administer.

To configure object access privileges for the SNMP Set feature:

1. Open the server.properties file.

- Windows: %NnmDataDir%nmsas\NNM\server.properties
- Linux: \$NnmDataDir/nmsas/NNM/server.properties
- 2. Add the following line:

```
permission.override.com.hp.nnm.SNMP SET=object-access-role
```

The following are the valid values for *object-access-role*:

```
com.hp.nnm.ADMIN
com.hp.nnm.LEVEL2
com.hp.nnm.LEVEL1
com.hp.nnm.GUEST
```

For example, to enable the **Object Administrator** and **Object Operator Level 2** object access privileges to use the SNMP Set feature, enter the following.

Example:

```
permission.override.com.hp.nnm.SNMP_SET=com.hp.nnm.ADMIN,com.hp.nnm.LEVEL
2
```

- 3. Include all the object access privileges for which you want to enable access.
- 4. Restart the NNMi ovjboss service by running the following commands on the NNMi management server:

```
ovstop ovjboss
ovstart
```

Changes to the server.properties file are read only at the time of ovjboss startup.

This section describes how to perform tasks.

21.8.1 Configuring NNMi to authenticate SNMPv3 traps for nodes that are either managed by using SNMPv2 or SNMPv1 or that are not discovered

Follow the steps in this section if NNMi is receiving SNMPv3 traps from nodes that meet either of the following criteria:

- The device is managed using SNMPv2 or SNMPv1.
- The device is not discovered by NNMi.

You can configure NNMi to add SNMPv3 engine IDs for these devices into the SNMPv3 cache.

By configuring NNMi this way, NNMi can authenticate and store these SNMPv3 traps.

To configure NNMi to receive and store SNMPv3 traps for nodes that are managed using SNMPv2 or SNMPv1 or that are not discovered:

1. In the NNMi console, navigate to **Configuration** > **Communication Settings**.

Configure the default, regions, or specific node settings level such that each inbound trap will have a corresponding configuration to use to authenticate the trap. For details, see *Configure Default SNMPv3 Settings* in NNMi Help.

Note

It is a good practice to use a region with included address ranges for your SNMPv3 nodes or configure a **Specific Node Setting** for each.

- 2. On the NNMi console, navigate to **Configuration** > **Incidents** > **Incident Configuration**.
- 3. Deselect Discard Unresolved SNMP Traps and Syslog Messages.

Once you have deselected **Discard Unresolved SNMP Traps and Syslog Messages**, NNMi will retain traps sent from nodes that it has not discovered.

- 4. Run the ovstop command on the NNMi management server.
- 5. Edit the following file:
 - Windows: %NNM_PROPS%\nms-communication.properties
 - Linux: \$NNM_PROPS/nms-communication.properties
- 6. Add the following line to the end of the file:

```
com.hp.nnm.snmp.engineid.file=file-pathfile.txt
```

The *file-pathfile.txt* entry is the full path and file name of the file that contains the devices.

With these configuration changes, NNMi will read the entries from this file into the SNMPv3 cache each time you restart the NNMi processes.

Important

On a Linux NNMi management server, the file path will be in the usual format, such as /var/opt/OV/etc.

On Windows NNMi management servers, use forward slashes for separators. For example, specify a file such as C:/temp/file.txt.

7. Save your changes.

8. Edit the *file-path*file.txt file.

a. Add IP addresses for the device, port, and engine ID, separating them with the comma.

b. Make the entry for each device on a separate line.

An engine ID is a series of hexadecimal bytes. NNMi ignores the character case but recognizes spaces.

Use the following examples as templates for creating your entries:

16.1.2.3,161,80 00 00 09 30 00 00 1f e9 a3 33 01 16.1.2.4,161,80 00 00 11 03 00 00 2d 51 99 30 00 1050:0000:0000:00005:0600:300c:326b, 161, 800000090300001f9ea33000 ff06::c3,161,80 00 00 09 03 00 00 1f 9A A3 30 00

a. Run the ovstart command on the NNMi management server to start NNMi and read the *file-path*file.txt file.

b. Check the Boot.log file to verify that NNMi read the file.

Verify that the file contains log messages similar to the following indicating that the file was read:

```
2012-10-17 14:44:44.876 INFO [NnmTrapService]Start: Populate engineIDs from file
2012-10-17 14:45:08.017 INFO [SnmpV3EngineIdCachePopulator]Successfully loaded 3 V
3
Engine IDs from file /temp/patch2/v3hosts.txt
```

If there was a failure mapping a node to a valid configuration, you will see a message similar to the following:

2012-10-17 14:45:03.485 WARNING [SnmpV3EngineIdCachePopulator]V3 Engine IDs: Could not resolve SNMPv3 configuration for 16.1.2.6

If you see a message similar to this, adjust the **Configuration** > **Communication Configuration** settings for the affected node.

🖹 Note

If you need to remove an entry from the cache as well as from the *file-pathfile.txt* file, it is best to remove the entry from the *file-pathfile.txt*, and then restart NNMi by running the following commands:

ovstop

ovstart

21.8.2 Block SNMPv1 or SNMPv2c Traps

Despite configuring device discovery to use only SNMPv3, some managed nodes may still try to send SNMPv1 or SNMPv2c traps to the NNMi management server. To prevent any SNMPv1 or SNMPv2c traps from reaching the NNMi

^{21.} Maintaining NNMi

management server, it is recommended that you configure NNMi to accept only SNMPv3 traps and block all SNMPv1 and SNMPv2c traps.

Note: Before completing this configuration procedure, make sure that NNMi is configured to discover your network with the SNMPv3 protocol.

- 1. Log on to the NNMi management server.
- 2. Run the following command:
 - Windows:

```
%NnmInstallDir%bin\nnmtrapconfig.ovpl -setProp disallowV1V2 -persist
```

• Linux:

```
/opt/OV/bin/nnmtrapconfig.ovpl -setProp disallowV1V2 -persist
```

- 3. Do one of the following:
 - Windows: Restart the NNM TrapReceiver service from the Services window.
 - Linux: Run the following commands:

- For distributions that uses systemd to manage services

```
/opt/OV/bin/nettrap stop
/opt/OV/bin/nettrap start
```

- Other distributions

```
/etc/init.d/nettrap stop
/etc/init.d/nettrap start
```

21.8.3 Configuring timeframes within which the Causal Engine stops accepting traps

When large areas of a network are unavailable at regular and predictable times, NNMi enables you to moderate the Causal Engine analysis load by inhibiting delivery of traps to the Causal Engine. To inhibit delivery of traps, you configure as an NNMi administrator the timeframes within which the NNMi Causal Engine is to stop accepting traps from the event system.

Important

This feature does not interfere with traps delivered to the NNMi console.

Traps that are delivered to the Causal Engine are used to trigger the State Poller to poll a node sooner than the schedule set by the State Poller polling policy. When you inhibit delivery of traps, NNMi must wait until the scheduled polling interval before obtaining updated information from the State Poller. In all cases, the NNMi Causal Engine reaches the same conclusion with or without traps by using status flows from the NNMi State Poller.

To configure times during which the Causal Engine is to stop accepting traps:

1. Create the following file:

- Windows: %NNM_PROPS%\nms-apa.properties
- Linux: \$NNM PROPS/nms-apa.properties

21. Maintaining NNMi

2. Add the following content to the created file:

PROPERTY NAME: com.hp.ov.nms.apa.trapGateSchedule

Use the following examples as guidelines:

In the following example, traps flow at midnight, are inhibited at 8:30 a.m., resume flowing at 10:00 a.m., and then are inhibited again at 4:30 p.m.:

com.hp.ov.nms.apa.trapGateSchedule = ENABLE APA TRAPS 08:30 10:00 16:30

In the following example, traps are inhibited at midnight, begin flowing at 8:30 a.m., are inhibited at 10:00 a.m., and then flow again at 4:30 p.m.:

com.hp.ov.nms.apa.trapGateSchedule = DISABLE_APA_TRAPS 08:30 10:00 16:30

- 3. Save your changes.
- 4. Restart NNMi by running the following commands:

ovstop ovstart

21.9 Blocking incidents using the trapFilter.conf file

Note the following if the number of incidents flowing through your NNMi management server reaches a rate that causes NNMi to block newly arriving incidents:

- NNMi will generate a TrapStorm incident, indicating that incidents are being blocked.
- NNMi might also generate a major health message indicating that the incident rate is high and incidents are being blocked.

Use either of the following methods to reduce the number of incidents:

• Use the nnmtrapd.conf file to block incidents from entering NNMi by reducing the volume of incident traffic.

Important

When you use the nnmtrapd.conf file approach, NNMi still uses these incidents to calculate the trap rate and to write to the trap binary store. By using the nnmtrapd.conf file approach, you only stop incidents from being created or stored in the database.

For details, see the *nnmtrapd.conf Reference Page*.

• Use trapFilter.conf to block incidents earlier in the NNMi event pipeline, preventing the blocked incidents from being analyzed for trap rate calculations and from being stored in the NNMi trap binary store.

Note

Adding device IP addresses or OIDs to the trapFilter.conf file enables you to block high-volume incidents and avoid problems caused by a high volume of incidents.

For details, see the *trapFilter.conf Reference Page* and the *nnmtrapconfig.ovpl Reference Page*.

21.10 Configuring character set encoding settings for NNMi

Depending on the locale configured for your NNMi management server, you might need to configure the source encoding NNMi will use to interpret SNMP OCTETSTRING data. To do this, edit the nms-jboss.properties file as follows:

- 1. Edit the following file:
 - Windows: %NNM_PROPS%\nms-jboss.properties
 - Linux: \$NNM_PROPS/nms-jboss.properties
- 2. Search for the text block containing the following line:

#!com.hp.nnm.sourceEncoding=UTF-8

3. Uncomment this line to read as follows:

```
com.hp.nnm.sourceEncoding=UTF-8
```

- 4. Modify the property value (UTF-8) shown in step 3 using the examples of comment statements shown in the nmsjboss.properties file.
- 5. Save the nms-jboss.properties file.
- 6. Restart NNMi by running the following commands:

ovstop ovstart

^{21.} Maintaining NNMi

21.11 Modifying MIB Browser Parameters

If you use the NNMi MIB browser (Action > MIB Information > Browse MIB menu) to obtain information about a node, and provide an optional SNMP community string for that node, the NNMi MIB browser uses MIB browser parameters located in the nms-ui.properties file for MIB Browser SNMP communication.

Important

If you do not provide a community string when using the MIB Browser, NNMi uses the Communication Configuration settings established for the node (if any). These settings are configured in the NNMi console using the Communications Settings view in the Configuration workspace. See Configuring Communication Protocol in the NNMi help for more information.

To modify the MIB Browser parameters in the nms-ui.properties file, follow these steps:

- 1. Edit the following file:
 - Windows: %NNM PROPS%\nms-ui.properties
 - Linux: \$NNM_PROPS/nms-ui.properties
- 2. Locate the text block containing the following line:

MIB Browser Parameters

3. Locate the MIB browser parameters located below # MIB Browser Parameters by searching for lines containing the following text:

mibbrowser

- 4. Modify the MIB browser parameters by following instructions within the nms-ui.properties file.
- 5. Save your changes.
- 6. Restart NNMi:

ovstop ovstart

21.12 Configuring NNMi to allow level 2 operators to delete nodes and incidents

By default, NNMi permits NNMi administrators to create, edit, or delete nodes or incidents in NNMi. You can configure accounts assigned to the NNMi Operator Level 2 (L2) User Group to have the ability to delete nodes or incidents. You can achieve this by using one of the following methods:

- (Recommended) Elevate the privileges of the required L2 users to delete the required nodes or incidents. This can be done using the NNMi web console. For more information, see the NNMi Admin help.
- Configure NNMi to globally enable L2 users to delete nodes or incidents. This can be done by overriding the default privileges by modifying certain NNMi property files.

Important

Use the override method only for global enablement. Once enabled, you cannot control L2 user access privileges in the NNMi web console.

To enable L2 users to edit or delete nodes, their associated incidents, or both, follow these steps:

- 1. Edit the following file:
 - Windows: %NNM_PROPS%\nms-topology.properties
 - Linux: \$NNM_PROPS/nms-topology.properties
- 2. Append the following lines as required:
 - To enable L2 users to delete nodes, append the following line:

permission.override.com.hp.nnm.DELETE_OBJECT=com.hp.nnm.ADMIN,com.hp.nnm.LEVEL2

• To enable L2 users to delete incidents, append the following line:

permission.override.com.hp.nnm.incident.DELETE=com.hp.nnm.ADMIN,com.hp.nnm.LEVEL
2

- 3. Save the file.
- 4. Restart NNMi.

ovstop
ovstart

When making file changes under HA, you must make the changes on both nodes in the cluster. For an NNMi that uses an HA configuration, if the change requires you to stop and restart the NNMi management server, you must place the nodes in maintenance mode before running the ovstop and ovstart commands.

21.13 Configuring NNMi to allow level 2 operators to edit maps

By default, NNMi permits only NNMi administrators to edit maps by creating, modifying, and deleting node groups. You can configure NNMi so that user accounts assigned to the NNMi Operator Level 2 user group also have this ability.

Following is the procedure for changing NNMi to permit user accounts assigned to the NNMi Operator Level 2 user group to create, modify, and delete node groups on nodes to which they have access:

1. Edit the following file:

- Windows: %NNM PROPS%\nms-ui.properties
- Linux: \$NNM_PROPS/nms-ui.properties
- 2. Search for the following text block and uncomment it:

```
#!com.hp.nnm.ui.level2MapEditing = true
```

- 3. Save your change.
- 4. Restart NNMi by running the following commands:

```
ovstop
ovstart
```

When making file changes under HA, you must make the changes on both nodes in the cluster. For an NNMi that uses an HA configuration, if the change requires you to stop and restart the NNMi management server, you must place the nodes in maintenance mode before running the ovstop and ovstart commands.

After completing step 1 through step 5, the NNMi console changes as follows:

- The Inventory > Node Group menu shows Create and Delete toolbar icons for the NNMi Operator Level 2.
- The toolbar on the Node Group form contains the Save and New and Delete Node Group buttons.
- The All Node Groups folder appears in the Topology Maps workspace. For details, see *About Workspaces* in NNMi online Help.
- For node group maps, the NNMi console contains the Save Map toolbar button and File > Save Map menu items.
- The behavior of the **Save Layout** action depends on the existence of a node group map setting for the node group map. If no node group map setting exists for a node group map, you must create one.

You can configure NNMi so that NNMi Operator Level 2 users have permission to create node group map settings:

- 1. From the NNMi console, open **Topology Maps** > **Node Group Overview**.
- 2. Double-click the Node Group icon of interest.

NNMi opens the node group map associated with the selected node group.

3. Open the node group map settings you want to modify:

Select File > Open Node Group Map Settings.

- 4. Set the Minimum NNMi Role to Save Layout to Operator Level 2.
- 5. Save your changes.

The NNMi operator level 2 user group can now create, edit, and delete node group map settings from a node group map view.

^{21.} Maintaining NNMi

21.14 Configuring NNMi to allow level 1 operators to run status polls and configuration polls

NNMi permits user accounts assigned to the NNMi Operator Level 2 user group to run status polls and configuration polls on nodes to which they have access. You must change the **Menu Item** configuration in the NNMi console as well as the object access privilege levels in the nms-topology.properties file for each poll.

If you need to change NNMi to permit user accounts assigned to the NNMi Operator Level 1 user group to run status polls and configuration polls, do the following:

- 1. Open the Configuration > User Interface > Menu Items > Status Poll form.
- 2. From the Menu Item Contexts tab, open the entry for each Required NNMi Role/Object Type item you need to change.
- 3. Change the value of the **Required NNMi Role** to **Operator Level 1** for each object type for which you want a level 1 operator to be able to run status polls.

This step enables the user accounts assigned to the NNMi Operator Level 1 user group to view the status poll action for the object type specified.

To change NNMi to permit user accounts assigned to the NNMi Operator Level 1 user group to view the **Configuration Poll** menu item, do the following:

- 1. Open the **Configuration** > **User Interface** > **Menu Items** > **Configuration Poll** form.
- 2. From the Menu Item Contexts tab, open the entry for each Required NNMi Role/Object Type item you need to change.
- 3. Change the value of the **Required NNMi Role** to **Operator Level 1** for each object type for which you want a level 1 operator to be able to run configuration polls.

This step enables the user accounts assigned to the NNMi Operator Level 1 user group to view the configuration poll action for the object type specified.

Editing the nms-topology.properties file, as shown in steps 7 through 10, permits user accounts assigned to the NNMi Operator Level 1 user group to run both status poll and configuration poll commands from the NNMi console. If you do not complete these steps, NNMi will display the status poll and configuration poll options in the **Actions** menu, but the user will see an error message when an attempt is made to run the status poll or configuration poll commands.

- 4. To change the level of access required for status polls and configuration polls (the required object access privilege levels), edit the following file:
 - Windows: %NNM_PROPS%\nms-topology.properties
 - Linux: \$NNM_PROPS/nms-topology.properties
- 5. Scroll to the end of the file and add the following line for the status poll change:

```
permission.override.com.hp.nnm.STATUS_POLL=com.hp.nnm.ADMIN,com.hp.nnm.LEVEL2,com.
hp.nnm.LEVEL1
```

6. Add the following line for the configuration poll change:

```
permission.override.com.hp.nnm.CONFIG_POLL=com.hp.nnm.ADMIN,com.hp.nnm.LEVEL2,com.
hp.nnm.LEVEL1
```

- 7. Save your changes.
- 8. Restart NNMi by running the following commands:

```
21. Maintaining NNMi
```

```
ovstop
ovstart
```

When making file changes under HA, you must make the changes on both nodes in the cluster. In the case of NNMi in an HA configuration, if the changes require that you stop and restart the NNMi management server, you must put the nodes in the maintenance mode before running the ovstop and ovstart commands.

21.15 Determining the original trap address from traps sent by a proxy SNMP gateway

Traps sent by a proxy SNMP gateway might not show the original trap address when NNMi is used in the default configuration. An administrator can configure NNMi to determine the original trap address.

Note the following:

- NNMi contains the cia.originaladdress custom incident attribute. NNMi determines the meaning of the cia.originaladdress attribute in conjunction with the com.hp.nnm.trapd.useUdpHeaderIpAddress property.
- The value of the com.hp.nnm.trapd.useUdpHeaderIpAddress parameter is false by default, so NNMi normally ignores the cia.originaladdress attribute.
- Once you have set the com.hp.nnm.trapd.useUdpHeaderIpAddress value to true, the cia.originaladdress attribute will provide the value of the SNMP agent address.

Setting the com.hp.nnm.trapd.useUdpHeaderIpAddress value to true is useful when you want to use the UDP header address as the source in NNMi but you still require access to the actual SNMP address of the managed device.



Important

When the com.hp.nnm.trapd.useUdpHeaderIpAddress attribute is false (the default setting), the cia.originaladdress and cia.address attributes both contain the same value.

To configure NNMi to determine the original trap address using the value of cia.originaladdress:

- 1. Edit the following file:
 - Windows: %NNM_PROPS%\nms-jboss.properties
 - Linux: \$NNM_PROPS/nms-jboss.properties
- 2. Search for the text block containing the following line:

#!com.hp.nnm.trapd.useUdpHeaderIpAddress=false

3. Uncomment and edit this line to read as follows:

com.hp.nnm.trapd.useUdpHeaderIpAddress=true

- 4. Save your changes.
- 5. Restart NNMi by running the following commands:

```
ovstop
ovstart
```

Important

When making file changes under High Availability (HA), you must make the changes on both nodes in the cluster. In the case of using NNMi in an HA configurations, if the changes require that you stop and restart the NNMi management server, you must put the nodes in the maintenance mode before

running the ovstop and ovstart commands. For details, see 19.6.1 Placing NNMi in maintenance mode.

NNMi uses the value of cia.originaladdress to determine the original trap address.

21.15.1 Trap address ordering

NNMi analyzes source addresses as follows:

• SNMPv1 and SNMPv2c traps in which the com.hp.nnm.trapd.useUdpHeaderIpAddress property is set to true use the following address order:

```
rfc3584TrapAddress (.1.3.6.1.6.3.18.1.3.0)
nnmTrapForwardingAddress (.1.3.6.1.4.1.11.2.17.2.19.1.1.3.0)
securityPackNotificationAddress (.1.3.6.1.4.1.99.12.45.2.1.0)
proxyOid (.1.3.6.1.4.1.11.2.17.5.1.0)
Source address in IP header
```

• SNMPv1 traps in which the com.hp.nnm.trapd.useUdpHeaderIpAddress property is set to false use the following address order:

```
rfc3584TrapAddress (.1.3.6.1.6.3.18.1.3.0)
nnmTrapForwardingAddress (.1.3.6.1.4.1.11.2.17.2.19.1.1.3.0)
securityPackNotificationAddress (.1.3.6.1.4.1.99.12.45.2.1.0)
proxyOid (.1.3.6.1.4.1.11.2.17.5.1.0)
agent-addr field in v1 trap
Source address in IP header
```

NNMi provides a standalone NmsTrapReceiver process for minimizing the loss of SNMP traps during a failover. NmsTrapReceiver runs on both the active and standby nodes.

21.16.1 Configuring the NmsTrapReceiver

NNMi provides the following user-configurable setting:

• trapReceiverJmsTTL

The trapReceiverJmsTTL option sets the maximum time that the TrapReceiver will cache traps. The default setting is 5 minutes. Data is lost if JBoss is down for longer than the amount of time that is set in trapReceiverJmsTTL.



Before you configure this setting, time a failover to determine a benchmark, and then set the trapReceiverJmsTTL to be double that amount of time.

For details about how to modify this setting, see the nnmtrapconfig.ovpl Reference Page.

Important

For proper operation, it is critical that the clocks are synchronized between the active and standby nodes. Otherwise, you might experience a large duplication or loss of traps.

For details, see the nnmtrapconfig.ovpl Reference Page.

21.16.2 Starting and stopping the NmsTrapReceiver process

The NmsTrapReceiver process is started automatically by the operating system (Linux: init.d nettrap or nettrap.service on systemd; Windows: HP NNM NmsTrapReceiver Service). The NmsTrapReceiver process is also started by ovstart if ovstart detects that the NmsTrapReceiver process is not running.

If you need to start or stop the NmsTrapReceiver manually, use the operating system service.

Important

The ovstart and ovstop commands only start and stop the JBoss pipeline for the processing of traps, not the remote trap server.

21.17 Configuring HTTPS-only communication with the NNMi console

The most effective method of preventing HTTP access to the NNMi console is to place the NNMi management server behind a firewall that permits only HTTPS access to the protected systems.

Using a firewall configuration to prevent HTTP access can cause problems for integrations that use Web services that support only HTTP to communicate with NNMi. See the documentation for the integrating product to determine whether it supports HTTPS.

For a less secure approach, redirect NNMi console access requests from the HTTP port to the HTTPS port by completing the following steps:

- 1. Edit the following file:
 - Windows: %NNM_PROPS%\nms-ui.properties
 - Linux: \$NNM_PROPS/nms-ui.properties
- 2. Search for the string https to locate the text block containing the following line:

#! com.hp.ov.nms.ui.https.only=false

3. Uncomment and edit this line to read as follows:

com.hp.ov.nms.ui.https.only=true

- 4. Save your changes.
- 5. Restart NNMi by running the following commands:

ovstop ovstart

Note

Setting this property to redirect HTTP requests to HTTPS for the NNMi console can cause problems with some applications that cross-launch back to NNMi. If you experience such problems, disable this HTTPS redirect.

21.18 Configuring NNMi to require encryption for remote access

The HTTP mode of communication can still be used even after installing and configuring NNMi to use HTTPS communication. To be able to restrict remote access to NNMi via HTTP, completely disable NNMi's HTTP mode of communication by the following instructions.

Before configuring NNMi to permit only encrypted remote access, make sure the global network management and other integrations support SSL. Configure them for SSL before configuring NNMi to permit only encrypted remote access.

Do not perform this task if you want to and are yet to configure the application failover cluster. After setting up the NNMi application failover cluster, you can complete these steps to disable HTTP and other unencrypted access.

To disable HTTP access from the network to NNMi, edit the server.properties file as follows:

- 1. Edit the following file (you will need to create it if it does not exist):
 - Windows: %NnmDataDir%nmsas\NNM\server.properties
 - Linux: \$NnmDataDir/nmsas/NNM/server.properties
- 2. Add the following four lines to the server.properties file:

```
nmsas.server.net.bind.address = 127.0.0.1
nmsas.server.net.bind.address.ssl = 0.0.0.0
nmsas.server.net.hostname = localhost
nmsas.server.net.hostname.ssl = ${com.hp.ov.nms.fqdn}
```

- 3. Save your changes.
- 4. Restart NNMi by running the following commands:

ovstop ovstart

^{21.} Maintaining NNMi

21.19 Configuring NNMi to preserve a previously supported varbind order

All SNMPv2 traps contain the sysUptime. 0 and snmpTrapOID. 0 OIDs as the first and second varbinds.

Important

When an SNMPv2 trap definition contains either or both of sysUptime.0 and snmpOID.0 as trap parameters, they might appear in NNMi as additional varbinds in positions other than first and second in the varbind list.

Prior to NNMi 10-10, NNMi removed all instances of the sysUpTime.0 and snmpTrapOID.0 OIDs from the varbind list. Starting with NNMi 10-10, NNMi retains these OIDs when they are part of the trap definition, but they might appear in positions other than first and second in the varbind list of received trap. This change in positioning might alter the varbind order for those traps that have either sysUpTime.0 or snmpTrapOID.0 as trap parameters.

In the following example, the first boldface varbind contains the value for snmpTrapOID.0 and the second boldface varbind contains the value for sysUpTime.0. As shown in this example, these varbinds appear as additional varbinds in positions other than first and second in the varbind list:

```
//0: SNMP MESSAGE (0x30): 115 bytes
//2: INTEGER VERSION (0x2) 1 bytes: 1 (SNMPv2C)
//5: OCTET-STR COMMUNITY (0x4) 6 bytes: "public"
//13: V2-TRAP-PDU (0xa7): 102 bytes
//15: INTEGER REQUEST-ID (0x2) 2 bytes: 18079
//19: INTEGER ERROR-STATUS (0x2) 1 bytes: noError(0)
//22: INTEGER ERROR-INDEX (0x2) 1 bytes: 0
//25: SEQUENCE VARBIND-LIST (0x30): 90 bytes
//27: SEQUENCE VARBIND (0x30): 13 bytes
//29: OBJ-ID (0x6) 8 bytes: .1.3.6.1.2.1.1.3.0
//39: TIMETICKS (0x43) 1 bytes: 9
//42: SEQUENCE VARBIND (0x30): 32 bytes
//44: OBJ-ID (0x6) 10 bytes: .1.3.6.1.6.3.1.1.4.1.0
//56: OBJ-ID (0x6) 18 bytes: .1.3.6.1.6.3.1.1.5.3.1.3.6.1.4.1.9.1.14
//76: SEQUENCE VARBIND (0x30): 14 bytes
//78: OBJ-ID (0x6) 9 bytes: .1.3.6.1.2.1.2.2.1.1
//89: INTEGER (0x2) 1 bytes: 92
//92: SEQUENCE VARBIND (0x30): 23 bytes
//94: OBJ-ID (0x6) 10 bytes: .1.3.6.1.6.3.1.1.4.3.0
//106: OBJ-ID (0x6) 9 bytes: .1.3.6.1.4.1.11.2.3.14
```

Note

Set the com.hp.nnm.events.preserveOldVarbindListOrder property to true only if you want NNMi to remove all instances of the sysUpTime.0 and snmpTrapOID.0 OIDs from the varbind list.

To retain the NNMi behavior that existed prior to NNMi 10-10, do the following:

- 1. Edit the following file:
 - Windows: %NNM_PROPS%\nms-jboss.properties
 - Linux: \$NNM_PROPS/nms-jboss.properties

21. Maintaining NNMi

2. Add the following line:

com.hp.nnm.events.preserveOldvarbindListOrder=true

3. Save your changes.

4. Restart NNMi by running the following commands:

ovstop ovstart

21.20 Configuring the auto-trim oldest SNMP trap incidents feature

To keep NNMi performing at a high level, NNMi drops incoming SNMP traps (including syslog messages) after storing a specific number of SNMP traps in its database. You can use the auto-trim oldest SNMP trap incidents feature to control the number of SNMP traps stored in the NNMi database and to retain important incoming SNMP traps.

Important

Until NNMi 12-00, NNMi did not trim root causes, and only trimmed SNMP trap incidents.

In NNMi 12-10 or later, either "all incidents" or "only SNMP trap incidents" can now be set. The default is "all incidents".

In NNMi 12-10 or later, the automatic trim function for SNMP trap incidents is enabled when a new installation is performed, and NNMi deletes old incidents from the database. (By default, an archive is not created.) In the case of a version upgrade, the previous setting is inherited.

com.hp.nnm.events.snmpTrapAutoTrimSetting

Refer to the following concrete example, and configure settings such as the settings to enable or disable the automatic trim function and to switch between "only SNMP traps" and "all incidents" as the trimming target.



To trim SNMP trap incidents manually from the NNMi database, use the nnmtrimincidents.ovpl command. For details, see the *nnmtrimincidents.ovpl Reference Page*.

21.20.1 Enabling the auto-trim oldest incidents feature (no incident archive)

If the number of incidents (including all types of incidents except SNMP traps and including syslog messages) in the database exceeds 50,000, perform the following procedure to use the automatic trim function for incidents to delete 10,000 incidents. Incidents are deleted in order starting from the oldest incident. Note that this procedure does not create an archive for incidents.

1. Edit the following file:

- Windows: %NNM_PROPS%\nms-jboss.properties
- Linux: \$NNM_PROPS/nms-jboss.properties
- 2. Look for the following line:

#!com.hp.nnm.events.snmpTrapAutoTrimStartPercentage=50

3. Uncomment and edit this line to read as follows:

com.hp.nnm.events.snmpTrapAutoTrimStartPercentage=50

4. Look for the following line:

#!com.hp.nnm.events.snmpTrapAutoTrimPercentageToDelete=25

```
21. Maintaining NNMi
```

5. Uncomment and edit this line to read as follows:

com.hp.nnm.events.snmpTrapAutoTrimPercentageToDelete=20

6. Look for the following line:

#!com.hp.nnm.events.snmpTrapAutoTrimSetting=Disabled

7. Uncomment and edit this line to read as follows:

com.hp.nnm.events.snmpTrapAutoTrimSetting=TrimOnly

If an archive is necessary, specify TrimAndArchive.

8. Find the row that includes the following character string.

com.hp.nnm.events.allowAutoTrimAllIncidents

9. Delete the comment symbol, and make the following correction.

com.hp.nnm.events.allowAutoTrimAllIncidents=true

10. Save your changes.

11. Restart NNMi by running the following commands:

```
ovstop
ovstart
```

The default value of com.hp.nnm.events.snmpTrapMaxStoreLimit (which sets the upper limit on the number of incidents) is 100,000. In this case, if the number of incidents in the database exceeds 50,000 (50% of the upper limit), 10,000 incidents are deleted in order, started from the oldest incident, and based on the following formula.

(com.hp.nnm.events.snmpTrapAutoTrimStartPercentage / 100) X com.hp.nnm.events.snmpTra pMaxStoreLimit X (com.hp.nnm.events.snmpTrapAutoTrimPercentageToDelete / 100) * "X" means multiplication.

21.20.2 Enabling the auto-trim oldest SNMP trap incidents feature (incident archive enabled)

Suppose you want to enable the auto-trim oldest SNMP trap incidents feature to trim 10,000 SNMP trap incidents (including syslog messages) once the number of SNMP trap incidents in the NNMi database exceeds 50,000. For this example, you want NNMi to archive the SNMP trap incidents before trimming them. Complete the following steps: SNMP trap incidents are deleted in order, starting from the oldest incident. Note that this procedure creates an archive for SNMP trap incidents.

1. Edit the following file:

- Windows: %NNM_PROPS%\nms-jboss.properties
- Linux: \$NNM_PROPS/nms-jboss.properties
- 2. Look for the line includes the following character string:

com.hp.nnm.events.snmpTrapAutoTrimStartPercentage

3. Uncomment and edit this line to read as follows:

com.hp.nnm.events.snmpTrapAutoTrimStartPercentage=50

```
21. Maintaining NNMi
```

4. Look for the line includes the following character string:

com.hp.nnm.events.snmpTrapAutoTrimPercentageToDelete

5. Uncomment and edit this line to read as follows:

com.hp.nnm.events.snmpTrapAutoTrimPercentageToDelete=20

6. Look for the line includes the following character string:

#!com.hp.nnm.events.snmpTrapAutoTrimSetting=Disabled

7. Uncomment and edit this line to read as follows:

com.hp.nnm.events.snmpTrapAutoTrimSetting=TrimAndArchive

8. Look for the line includes the following character string:

com.hp.nnm.events.allowAutoTrimAllIncidents

9. Uncomment and edit this line to read as follows:

```
com.hp.nnm.events.allowAutoTrimAllIncidents=false
```

10. Save your changes.

11. Restart NNMi by running the following commands:

```
ovstop
ovstart
```

The default value of com.hp.nnm.events.snmpTrapMaxStoreLimit (which sets the upper limit on the number of incidents) is 100,000. In this case, if the number of SNMP trap incidents (including syslog messages) in the database exceeds 50,000 (50% of the upper limit), 10,000 SNMP trap incidents are deleted in order, started from the oldest incident, and based on the following formula.

```
(com.hp.nnm.events.snmpTrapAutoTrimStartPercentage / 100) X com.hp.nnm.events.snmpTra
pMaxStoreLimit X (com.hp.nnm.events.snmpTrapAutoTrimPercentageToDelete / 100)
* "X" means multiplication.
```

The deleted incidents are archived in the following file:

- Windows: %NNM TMP%\incidentArchive."date".csv.gz
- Linux: \$NNM TMP/incidentArchive."date".csv.gz

The archive file name is fixed until the NNMi service is restarted and incidents are added to the same file.

21.20.3 Rotate archive files

In default setting, there is not upper limit of the file size of the archive file created by the auto-trim feature. You can rotate the archive file in specific file size and save disk usage. Suppose you want to rotate the archive files 3 times when the size reaches 10MB. Complete the following steps:

1. Edit the following file:

- Windows: %NNM_PROPS\nms-jboss.properties
- Linux: \$NNM_PROPS/nms-jboss.properties

21. Maintaining NNMi

2. Locate the text block containing the following line:

#!com.hp.nnm.events.autoTrimArchiveRotationEnabled=false

3. Uncomment and edit the line to read as follows:

com.hp.nnm.events.autoTrimArchiveRotationEnabled=true

4. Locate the text block containing the following line:

#!com.hp.nnm.events.autoTrimArchiveRotationArchiveSize=128

5. Uncomment and edit the line to read as follows:

com.hp.nnm.events.autoTrimArchiveRotationArchiveSize=10

6. Locate the text block containing the following line:

#!com.hp.nnm.events.autoTrimArchiveRotationRotateNumber=5

7. Uncomment and edit the line to read as follows:

com.hp.nnm.events.autoTrimArchiveRotationRotateNumber=3

8. Restart the NNMi management server:

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

 $b. \ Run \ the \ \mbox{ovstart} \ command \ on \ the \ NNMi \ management \ server.$

Important

When making file changes under High Availability (HA), you need to make the changes on both nodes in the cluster. For NNMi using HA configurations, if the change requires you to stop and restart the NNMi management server, you must put the nodes in maintenance mode before running the ovstop and ovstart commands.

With this settings, trimmed incidents are archived into the following file. Note that the file name doesn't contain the time stamp.

- Windows: %NNM_TMP\incidentArchive.csv.gz
- Linux: \$NNM_TMP/incidentArchive.csv.gz

When the file size of the archive file reached the specified value in MB (10 in this example), the archive file is renamed to "incidentArchive.csv.gz.1". The new incidents are archived into the new incidentArchive.csv.gz.

"incidentArchive.csv.gz.<n>" is renamed to "incidentArchive.csv.gz.<n+1>" by the rotation. If the value of "n" in the archive file name is the same as the specified value (3 in this example), the file is removed by the rotation, instead of being renamed.

21.20.4 Changing the maximum number of SNMP trap incidents to be saved

Depending on whether you need to keep SNMP trap incidents for a long period of time, you can change the maximum number of SNMP trap incidents to be saved in the database.
Important

By default, NNMi begins dropping SNMP traps (including syslog messages) after the number of SNMP trap incidents in its database reaches 100,000. Setting this limit to a higher value is not recommended, as doing so can cause NNMi performance degradation. Carefully evaluate the situation before changing the maximum number.

(1) Changing the maximum number to less than 100,000

To change the maximum number to 50,000:

- 1. Edit the following file:
 - Windows: %NNM_PROPS%\nms-jboss.properties
 - Linux: \$NNM_PROPS/nms-jboss.properties
- 2. Look for the following line:

#!com.hp.nnm.events.snmpTrapMaxStoreLimit=100000

3. Uncomment and edit this line to read as follows:

com.hp.nnm.events.snmpTrapMaxStoreLimit=50000

- 4. Save your changes.
- 5. Restart NNMi by running the following commands:

```
ovstop
ovstart
```

(2) Changing the maximum number to more than 100,000

To change the maximum number to 200,000:

- 1. Edit the following file:
 - Windows: %NNM PROPS%\nms-jboss.properties
 - Linux: \$NNM_PROPS/nms-jboss.properties
- 2. Look for the following line:

#!com.hp.nnm.events.snmpTrapEnforce100KLimit=true

3. Uncomment and edit this line to read as follows, and save the file:

com.hp.nnm.events.snmpTrapEnforce100KLimit=false

4. Edit the following file:

- Windows: %NNM_PROPS%\nms-jboss.properties
- Linux: \$NNM_PROPS/nms-jboss.properties
- 5. Look for the following line:

```
#!com.hp.nnm.events.snmpTrapMaxStoreLimit=100000
```

6. Uncomment and edit this line to read as follows:

```
21. Maintaining NNMi
```

com.hp.nnm.events.snmpTrapMaxStoreLimit=200000

- 7. Save your changes.
- 8. Restart NNMi by running the following commands:

```
ovstop
ovstart
```

The default value of com.hp.nnm.events.snmpTrapMaxStoreLimit (which sets the upper limit on the number of incidents) is 100,000. In this case, if the number of incidents in the database exceeds 50,000 (50% of the upper limit), 10,000 incidents are deleted in order, started from the oldest incident, and based on the following formula.

21.20.5 Monitoring the auto-trim oldest SNMP trap incidents feature

From the NNMi console, click **Help**, choose **System Information**, and then **Health** to check whether messages have been output regarding the health of the auto-trim oldest SNMP trap incidents feature. NNMi also generates the following alarms related to the auto-trim oldest SNMP trap incidents feature:

- NNMi generates SnmpTrapLimitCritical when the number of SNMP trap incidents stored in the database (including syslog messages) reaches 100% of the com.hp.nnm.events.snmpTrapMaxStoreLimit value.
- NNMi generates SnmpTrapLimitMajor when the number of SNMP trap incidents stored in the database (including syslog messages) reaches 95% of the com.hp.nnm.events.snmpTrapMaxStoreLimit value.
- NNMi generates SnmpTrapLimitWarning when the number of SNMP trap incidents stored in the database (including syslog messages) reaches 90% of the com.hp.nnm.events.snmpTrapMaxStoreLimit value.

Important

When making file changes under High Availability (HA), you must make the changes on both nodes in the cluster. In the case of NNMi in an HA configuration, if the change requires that you stop and restart the NNMi management server, you must put the nodes in the maintenance mode before running the ovstop and ovstart commands. For details, see 19.6.1 Placing NNMi in maintenance mode.

21.20.6 Disabling the auto-trim oldest SNMP trap incidents feature

To disable the auto-trim oldest SNMP trap incidents feature, complete the following steps:

- 1. Edit the following file:
 - Windows: %NNM PROPS%\nms-jboss.properties
 - Linux: \$NNM PROPS/nms-jboss.properties
- 2. Locate the line containing the following property:

com.hp.nnm.events.snmpTrapAutoTrimSetting

3. Edit this line to read as follows:

com.hp.nnm.events.snmpTrapAutoTrimSetting=Disabled

4. Save your changes.

```
21. Maintaining NNMi
```

5. Restart NNMi by running the following commands:

ovstop ovstart

21.21 Modifying NNMi normalization properties

NNMi stores both host names and node names in case-sensitive form. This means that all searches, sorts, and filters that the NNMi console provides return case-sensitive results. If the DNS servers you use return a variety of case-preserving node names and host names, including all-uppercase, all-lowercase, and a mixture of uppercase and lowercase, this can cause less-than-optimal results.

You can change several NNMi normalization properties to meet your specific needs. A good practice is to make these changes before seeding NNMi for its initial discovery.

We recommend that you adjust the settings in this section during deployment, but before running the initial discovery.

If you run an initial discovery, and then decide to change the normalization properties later, you can run the nnmnoderediscover.ovpl -all script to initiate a full discovery. For details, see the *nnmnoderediscover.ovpl Reference Page*.

You can change the following properties:

- Normalize discovered node names to UPPERCASE, LOWERCASE, or OFF.
- Normalize discovered host names to UPPERCASE, LOWERCASE, or OFF.

To change normalization properties follow these steps:

1. Edit the following file:

- Windows: %NNM_PROPS%\nms-topology.properties
- Linux: \$NNM_PROPS/nms-topology.properties
- 2. To configure NNMi to normalize discovered names, look for a line that resembles the following:

#!com.hp.ov.nms.topo.NAME_NORMALIZATION=OFF

a. Uncomment the property (to uncomment a property, remove the # ! characters from the beginning of the line):

com.hp.ov.nms.topo.NAME_NORMALIZATION=OFF

b. Change OFF to LOWERCASE or UPPERCASE.

c. Save your changes.

3. To configure NNMi to normalize discovered host names, look for a line that resembles the following:

#!com.hp.ov.nms.topo.HOSTNAME_NORMALIZATION=OFF

a. Uncomment the property:

com.hp.ov.nms.topo.HOSTNAME_NORMALIZATION=OFF

To uncomment a property, remove the #! characters from the beginning of the line.

b. Change OFF to LOWERCASE or UPPERCASE.

c. Save your changes.

4. Restart NNMi by running the following commands:

ovstop	
ovstart	

21.21.1 Changing normalization properties following an initial discovery

Changing normalization properties following an initial discovery causes NNMi to be inconsistent with the property changes until the next discovery. To remedy this, run the nnmnoderediscover.ovpl-all script to initiate a full discovery after changing NNMi normalization properties.

^{21.} Maintaining NNMi

21.22 Modifying the database port

You might want to configure NNMi to use a different port for the database. To do so, follow these steps:

1. Edit the following file:

- Windows: %NNM_CONF%\nnm\props\nms-local.properties
- Linux: \$NNM_CONF/nnm/props/nms-local.properties
- 2. Search for a line that resembles the following:

#!com.hp.ov.nms.postgres.port=5432

3. Uncomment this property:

com.hp.ov.nms.postgres.port=5432

To uncomment a property, remove the characters #! from the beginning of its line.

- 4. Change the existing value to the new port number.
- 5. Save your changes.
- 6. Restart NNMi by running the following commands:

```
ovstop
ovstart
```

21.23 NNMi self monitoring

NNMi performs self-monitoring checks, including checks of memory, CPU, and disk resources. NNMi generates an incident when the NNMi management server becomes low on a resource or detects a serious condition.

To view NNMi health information, use one of the following methods:

- From the NNMi console, click Help, choose System Information, and then click the Health tab.
- Run the nnmhealth.ovpl script.

NNMi opens a status message at the bottom of the NNMi console and at the top of forms when it detects a self-monitoring health exception. You can disable this warning message by completing the following steps:

- 1. Edit the following file:
 - Windows: %NNM_PROPS%\nms-ui.properties
 - Linux: \$NNM PROPS/nms-ui.properties
- 2. Search for the text block containing the following line:

#!com.hp.nms.ui.health.disablewarning=false

3. Uncomment and edit this line to read as follows:

com.hp.nms.ui.health.disablewarning=true

- 4. Change the existing value to the new port number.
- 5. Save your changes.
- 6. Restart NNMi by running the following commands:

ovstop ovstart

^{21.} Maintaining NNMi

21.24 Suppressing the use of discovery protocols for specific nodes

NNMi uses several protocols to discover Layer 2 connectivity between and among network devices. There are many defined discovery protocols; for example, *Link Layer Discovery Protocol* (LLDP) is an industry standard protocol. There are also many vendor-specific protocols, such as *Cisco Discovery Protocol* (CDP) for Cisco devices.

You can configure NNMi to suppress discovery protocol collections for devices you specify. There are special circumstances, such as those described below, which might be remedied by suppressing discovery protocol collections.

Enterasys devices:

Using SNMP to collect information from the *Enterasys Discovery Protocol* (EnDP) and LLDP tables on some Enterasys devices might cause issues with NNMi running out of memory. You can prevent this by configuring NNMi to skip EnDP and LLDP processing on these devices. To do this, add the management addresses of the devices to the disco.SkipXdpProcessing file. For details, see 21.24.1 Suppressing the use of discovery protocol collections.

New operating system versions on some Enterasys devices support the set snmp timefilter break command. On such Enterasys devices, run the set snmp timefilter break command. When you use this command to configure such a device, you do not need to list the device in the disco.SkipXdpProcessing file.

Nortel devices:

Many Nortel devices use the *SynOptics Network Management Protocol* (SONMP) to discover Layer 2 layout and connectivity. Some of these devices use the same MAC address on multiple interfaces, which does not work well with this protocol. You might experience this problem if two interconnected Nortel devices show a Layer 2 connection between the wrong set of interfaces and the connection shows a connection source of SONMP.

For this example, it is best to configure NNMi to not use the SONMP protocol to derive Layer 2 connections for the devices shown as participating in the wrong connection. To do this, add the management address of the two devices to the disco.SkipXdpProcessing file. For details, see 21.24.1 Suppressing the use of discovery protocol collections.

21.24.1 Suppressing the use of discovery protocol collections

To suppress the use of discovery protocol collections, follow these steps:

1. Create the following file:

- Windows: %NnmDataDir%shared\nnm\conf\disco\disco.SkipXdpProcessing
- Linux: \$NnmDataDir/shared/nnm/conf/disco/disco.SkipXdpProcessing The disco.SkipXdpProcessing file is case-sensitive.
- 2. Add to the disco.SkipXdpProcessing file the management address for each device for which protocol collection is to be suppressed.

For details, see the disco.SkipXdpProcessing Reference Page.

3. Restart NNMi by running the following commands:

```
ovstop
ovstart
```



Important

Suppressing discovery protocol processing of a node or nodes might cause some inaccuracies in the Layer 2 layout of the managed network.

The ovjboss service reads the disco. SkipXdpProcessing file at startup. If you make any changes after starting NNMi, restart NNMi as shown in this step.

If you ran the set snmp timefilter break command on any Enterasys devices, remove their device addresses from the disco.SkipXdpProcessing file, and then restart NNMi as shown in this step. NNMi opens more accurate Layer 2 maps when it uses discovery protocols.

For details, see the *disco.SkipXdpProcessing Reference Page*.

21.25 Configuring actions for secondary root cause management events

By default, NNMi does not run actions on secondary root cause management events.

This helps prevent unnecessary generation of actions. For example, if NNMi detects an InterfaceDown incident, and soon after it determines that the corresponding card is down, then, if dampening is used, the CardDown incident becomes the root cause and the InterfaceDown incident is downgraded to secondary.

In this case, an action on the InterfaceDown incident is not desired because such an action will be now applied to the new root cause (CardDown).

To enable actions for secondary root cause management events, do the following:

- 1. Edit the following file:
 - Windows: %NNM PROPS%\nms-jboss.properties
 - Linux: \$NNM PROPS/nms-jboss.properties
- 2. Search for the text block containing the following line:

#!com.hp.nnm.events.action.runActionOnSecRootCauseMgmtEvent=false

3. Uncomment and edit this line to read as follows:

com.hp.nnm.events.action.runActionOnSecRootCauseMgmtEvent=true

- 4. Save the nms-jboss.properties file.
- 5. Restart NNMi by running the following commands:

ovstop ovstart

21. Maintaining NNMi

21.26 Scheduling outages

NNMi enables you to use the nnmscheduledoutage.ovpl command to schedule outages for any set of nodes. For example, you might want to schedule an outage for weekly maintenance on a set of routers, or perhaps to replace the power supply for an individual node.

For details, see the *nnmscheduledoutage.ovpl Reference Page*.



For details about using NNMi to schedule outages, see NNMi Help.

21. Maintaining NNMi

NNMi includes the following physical sensors and node sensors, which can be monitored as an aid in determining status.

Physical sensor	Propagates status to physical component by default?	Node sensor	Propagates status to node by default?
FAN	Yes	СРИ	No
POWER_SUPPLY	Yes	MEMORY	Yes
TEMPERATURE	No	BUFFERS	No
VOLTAGE	No	DISK_SPACE	No
BACK_PLANE	Yes		

Table 21-2: Physical sensors and node sensors



Important

By default, FAN, POWER SUPPLY, BACK PLANE, and MEMORY propagate their status to the physical component level. For example, if a fan has a red status indicator, its corresponding physical component (chassis) receives a status indicator of yellow. A user viewing the status of a chassis in this case would be alerted to the fact that a component of that chassis has experienced some sort of failure.

21.27.1 Configuring physical sensor status

You can configure whether a physical sensor propagates its status to the physical component level (for example, the chassis) by following the steps in the following subsections.

(1) Propagating physical sensor status to a physical component

- 1. If not already present, create a new properties file with the name nnm-apa.properties in the following directory:
 - Windows: %NnmDataDir%shared\nnm\conf\props
 - Linux: \$NnmDataDir/shared/nnm/conf/props
- 2. Use a text editor to set the following text in this properties file:

com.hp.ov.nms.apa.PhysSensorPropagateToPhysicalComponentStatus type=true

type is the physical sensor. For details, see 21.27 Configuring sensor status.

- 3. Save the properties file.
- 4. Restart NNMi by running the following commands:

```
ovstop
ovstart
```

Important

When making file changes under High Availability (HA), you must make the changes on both nodes in the cluster. In the case of NNMi in an HA configurations, if the changes require that you stop and restart the NNMi management server, you must put the nodes in the maintenance mode before running the ovstop and ovstart commands. For details, see 19.6.1 Placing NNMi in maintenance mode.

(2) Configuring physical sensor status to not propagate to the physical component

- 1. If not already present, create a new properties file with the name nnm-apa.properties in the following directory:
 - Windows: %NnmDataDir%shared\nnm\conf\props
 - Linux: \$NnmDataDir/shared/nnm/conf/props
- 2. Use a text editor to include the following text in this properties file:

com.hp.ov.nms.apa.PhysSensorNoPropagateToPhysicalComponentStatus_type=true

type is the physical sensor. For details, see 21.27 Configuring sensor status.

- 3. Save the properties file.
- 4. Restart NNMi by running the following commands:

```
ovstop
ovstart
```

Important

When making file changes under High Availability (HA), you must make the changes on both nodes in the cluster. In the case of NNMi in an HA configuration, if the changes require that you stop and restart the NNMi management server, you must put the nodes in the maintenance mode before running the ovstop and ovstart commands. For details, see 19.6.1 Placing NNMi in maintenance mode.

(3) Overriding physical sensor status values

By default, the three sensor status values None, Warning, and Unavailable are mapped to the Normal status by the Causal Engine. You can override these default status mappings so that None, Warning, and Unavailable map to Critical.

To override physical sensor status values:

- 1. If not already present, create a new properties file with the name nnm-apa.properties in the following directory:
 - Windows: %NnmDataDir%shared\nnm\conf\props
 - Linux: \$NnmDataDir/shared/nnm/conf/props
- 2. Use a text editor to include one, two, or all three of the following lines, as applicable, in this properties file:

```
com.hp.ov.nms.apa.PhysSensorValueReMappedToDown_NONE=true
com.hp.ov.nms.apa.PhysSensorValueReMappedToDown_Warning=true
com.hp.ov.nms.apa.PhysSensorValueReMappedToDown_Unavailable=true
```

```
21. Maintaining NNMi
```

3. Save the properties file.

4. Restart NNMi by running the following commands:

```
ovstop
ovstart
```

Important

- You can map the Unavailable status to Unpolled status (because Unavailable means that the measurement facility is not available). Often, this situation might occur because the sensor has become nonfunctional, rather than because the component has become nonfunctional. To map Unavailable to Unpolled, use the same procedure as described above, except use the following text in step 2: com.hp.ov.nms.apa.PhysicalSensorValueReMappedToUnpolled_Unavailable= true
- When making file changes under High Availability (HA), you must make the changes on both nodes in the cluster. In the case of NNMi in an HA configuration, if the changes require that you stop and restart the NNMi management server, you must put the nodes in the maintenance mode before running the ovstop and ovstart commands. For details, see 19.6.1 Placing NNMi in maintenance mode.

21.27.2 Configuring node sensor status

You can configure whether a node sensor propagates its status to the node level by following the steps in the following subsections.

(1) Propagating node sensor status to a node

- 1. If not already present, create a new properties file with the name nnm-apa.properties in the following directory:
 - Windows: %NnmDataDir%shared\nnm\conf\props
 - Linux: \$NnmDataDir/shared/nnm/conf/props
- 2. Use a text editor to include the following text in this properties file:

com.hp.ov.nms.apa.NodeSensorPropagateToNodeStatus_type=true

type is the physical sensor. For details, see 21.27 Configuring sensor status.

- 3. Save the properties file.
- 4. Restart NNMi by running the following commands:

```
ovstop
ovstart
```

Important

When making file changes under High Availability (HA), you must make the changes on both nodes in the cluster. In the case of NNMi in an HA configuration, if the changes require that you stop and restart the NNMi management server, you must put the nodes in the maintenance mode before running the ovstop and ovstart commands. For details, see 19.6.1 Placing NNMi in maintenance mode.

(2) Configuring a node sensor's status to not propagate to the node

- 1. If not already present, create a new properties file with the name nnm-apa.properties in the following directory:
 - Windows: %NnmDataDir%shared\nnm\conf\props
 - Linux: \$NnmDataDir/shared/nnm/conf/props
- 2. Use a text editor to include the following text in this properties file:

com.hp.ov.nms.apa.NodeSensorNoPropagateToNodeStatus_type=true

type is the physical sensor. For details, see 21.27 Configuring sensor status.

- 3. Save the properties file.
- 4. Restart NNMi by running the following commands:

ovstop ovstart

Important

When making file changes under High Availability (HA), you must make the changes on both nodes in the cluster. In the case of NNMi in an HA configuration, if the changes require that you stop and restart the NNMi management server, you must put the nodes in the maintenance mode before running the ovstop and ovstart commands. For details, see 19.6.1 Placing NNMi in maintenance mode.



Changing the NNMi Management Server

You can duplicate the NNMi configuration on another system, such as 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.

22.1 Best practices for preparing the NNMi configuration to be moved

The following best practices are effective for moving the NNMi configuration to a different system:

- If the node group configuration uses host names to identify managed nodes, the production and test NNMi management servers must use the same DNS servers. In the case where 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
 - Menu
 - Menu item
 - Custom correlation configuration
 - Icon
 - MIB expression
 - Trap log configuration

^{22.} Changing the NNMi Management Server

22.2 Moving the NNMi configuration and database

To move the NNMi configuration and the database, such as from a test system to a production system, make a complete backup of all NNMi data on the source (test) system, and then restore the backup to the target (production) system. To ensure that no changes are made to the NNMi database after the backup is made, stop all NNMi processes and create an offline backup.

Example:

nnmbackup.ovpl -type offline -scope all -target nnmi backups\offline

Verify that the items listed in 20.3.2 Different-system restore are the same in both systems, and then execute a command similar to the following example.

Example:

nnmrestore.ovpl -source nnmi backups\offline\newest backup



Important

NNMi uses the same SSL certificate for accessing the database and for supporting HTTPS access to the NNMi console. The certificate for accessing the database was created when the NNMi processes first started on the source system. This certificate is included in the backup and restore data. Without this certificate, NNMi cannot access the database from the target system.

However, for HTTPS access to the NNMi console, the SSL certificate must be generated on the target system. Because the current implementation of JBoss does not support certificate merging, NNMi does not support HTTPS access to the NNMi console on a system that was set up by restoring data from a different system. If the target system must support HTTPS access to the NNMi console, use the procedure described in 22.3 Moving the NNMi configuration, and then begin data collection anew on the target system.

^{22.} Changing the NNMi Management Server

22.3 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.

Important

Do not edit a file exported with the nnmconfigexport.ovpl script before using the nnmconfigimport.ovpl script to import the file.

For details about these commands, see the appropriate reference pages.



- The nnmconfigexport.ovpl command does not retain SNMPv3 credentials. For details, see the *nnmconfigexport.ovpl Reference Page*.
- You can move only the NNMi configuration. Hitachi does not support moving topology or incident data from one NNMi management server to another NNMi management server.

^{22.} Changing the NNMi Management Server

22.4 Changing the IP address of a stand-alone NNMi management server

If you need to change the IP address of an NNMi management server, do so after stopping NNMi. After the address change, apply the license key that corresponds to the changed address and then start NNMi.

^{22.} Changing the NNMi Management Server

22.5 Changing the host name or domain name of an NNMi management server

Important

If the NNMi management server participates in NNMi application failover or participates global network management feature or contact your support representative for assistance.

To change the NNMi management server's host name, domain name, or both, set NNMi to use the NNMi management server's new fully qualified domain name (FQDN).

Example:

nnmsetofficialfqdn.ovpl -force newnnmi.servers.example.com

For details, see the nnmsetofficialfqdn.ovpl Reference Page.



Important

The FQDN is a host name combined with a domain name. If you change either of these, you are changing the NNMi management server's FQDN. SSL certificates are always linked to the FQDN. The common name (CN) field in the certificate must match the server FQDN. Therefore, if you change the FQDN, you must obtain a new SSL certificate with a matching CN. The nnmsetofficialfqdn.ovpl command updates the NNMi management server's FQDN and it also creates a new self-signed certificate, which matches the new FQDN. However, if you are using CA certificates, you must generate the new CA certificate. For details, see 10.3.2 Generating a CA-Signed Certificate.

If you change the IP address of the NNMi management server (regardless of whether the FQDN changes), you must obtain a new license. For details, see 22.4 Changing the IP address of a stand-alone NNMi management server.

22. Changing the NNMi Management Server



NNMi Security

This chapter explains security issues.

23.1 Providing a password for embedded database tools

NNMi requires a password to run embedded database tools (such as psql). Initially, NNMi provides a default password, which the user must change by running the nnmchangeembdbpw.ovpl script. You must be logged in as an administrator on a Windows system or as root on a Linux system to run the nnmchangeembdbpw.ovpl script. For details, see the *nnmchangeembdbpw.ovpl Reference Page*.

In an HA environment, you run the nnmchangeembdbpw.ovpl script on the primary cluster node only. The application automatically copies the password to the secondary cluster node; no additional action is required.

23.2 Configure TLS Protocols

By default, NNMi supports the TLSv1.2 protocol for HTTPS communication.

It is recommended that NNMi use only TLSv1.2 unless older, less secure, protocols are necessary for supporting legacy clients.

To configure NNMi to use protocols other than TLSv1.2, follow these steps:

- 1. Log on to the NNMi management server.
- 2. Open the following file with a text editor:
 - Windows: %NnmDataDir%nmsas\NNM\server.properties
 - Linux: /var/opt/OV/nmsas/NNM/server.properties
- 3. Adding or updating the com.hp.ov.nms.ssl.PROTOCOLS property with a comma-separated list of the protocols that you want to use.

For example, if you want to use the TLSv1, TLSv1.1, and TLSv1.2 protocols, make sure the following line exists in the server.properties file:

com.hp.ov.nms.ssl.PROTOCOLS=TLSv1.2,TLSv1.1,TLSv1

- 4. Restart the NNMi processes by running the following commands:
 - Windows:

```
%NnmInstallDir%bin\ovstop -c
%NnmInstallDir%bin\ovstart -c
```

• Linux:

```
/opt/OV/bin/ovstop -c
/opt/OV/bin/ovstart -c
```

23.3 NNMi data encryption

NNMi incorporates data encryption in many areas of the product.

Example:

NNMi stores passwords for user accounts in the NNMi database in encrypted form.

23.3.1 Encryption and user account passwords

Important

This information does not apply to the Lightweight Directory Access Protocol (LDAP) or to Common Access Card (CAC) accounts.

NNMi user accounts created from the NNMi console are stored in the NNMi database. The passwords for these users are hashed and also stored in the database.

When a user signs into the NNMi console, or uses a command line interface (CLI) tool, the password the user provides is hashed and compared to the hashed value stored in the database. If the user provides the correct password, these two hashed strings will match, authenticating the user.

Earlier versions of NNMi (10-50 or earlier) used encryption algorithms for hashing user passwords; these algorithms are now considered outdated. NNMi 11-00 uses a stronger algorithm for user account passwords. However, because hashing constitutes one-way encryption, it is not possible to decrypt and then re-encrypt user passwords during an upgrade from NNMi 10-50 to 11-00.

During an upgrade, all existing users still have their passwords stored in the database using the legacy encryption algorithm. However, when a user whose password has been hashed using the legacy algorithm logs on successfully, the password the user has provided is re-encrypted automatically using the new hash algorithm specified in the crypto configuration file.

This means all passwords are updated to the new algorithm slowly over time, as each user logs in for the first time after the upgrade. The same will be true if the crypto configuration is changed in the future. Each user's password will be upgraded to the new hash algorithm at the time of the user's next successful logon.

Important

- Upgrading user passwords depends on the presence of the earlier legacy algorithm (for example, MD5) listed in the <allowed> block. For this reason, you must keep the earlier legacy algorithm listed in the <allowed> block until all passwords have been migrated.
- If the earlier legacy algorithm is not kept in the <allowed> block, it will not be possible to re-hash the existing passwords that exist in hashed form in the database. In such a case, those users will not be able to log on, and NNMi will not be able to use the new algorithm to re-encrypt their passwords.
- Once the earlier legacy algorithm has been removed from the <allowed> block, the administrator either must delete and re-create the affected users or must reset the passwords for the users whose passwords were encrypted with the earlier legacy algorithm.

Use the following command to determine if a user's password can be used with an algorithm listed in the crypto configuration file or is still encrypted with an earlier legacy algorithm that is no longer specified in the crypto configuration file:

nnmsecurity.ovpl-listUserAccounts legacy

For details, see the nnmsecurity.ovpl Reference Page.

Part 7: Migration



Upgrading from NNMi Version 9, 10, or 11

This chapter explains several possible version upgrade scenarios.

For details about upgrading from NNM version 8 or earlier to NNMi, see <u>26. Upgrading from NNM</u> <u>Version 8 or Earlier</u>. If you plan to upgrade an NNMi management server of version 9, 10, or 11 that is running in an NNMi application failover configuration, the supported upgrade procedure requires you to unconfigure application failover temporarily, upgrade each of the NNMi management servers, and then reconfigure application failover.

To upgrade NNMi version 9, 10, or 11 that is running in a high-availability (HA) cluster, see the *Release Notes*.

24.1.1 Upgrading NNMi management servers from version 12-10

This subsection describes how to upgrade an NNMi management server that is running NNMi version 12-10.



Before upgrading an NNMi management server, see 1. Preinstallation Checklists.

1. Run the nnmbackup.ovpl script to back up the NNMi management server.

You perform this step only as a precaution, because you would use this backup only in the unlikely event of a failed migration. For details, see the *nnmbackup.ovpl Reference Page*.

- 2. Follow the procedure described in 2. Installing and Uninstalling NNMi to install a version 12-50 NNMi management server.
- 3. Verify that the information from the NNMi management server migrated successfully.

24.1.2 Upgrading NNMi management servers from version 9, 10, or 11

This subsection describes how to upgrade NNMi management servers running NNMi version 9, 10, or 11.

1. Follow the procedure described in the setup guide, [#] and the Release Notes for NNMi 12-10 to upgrade an NNMi management server to an NNMi management server running NNMi 12-10.

#: The setup guide refers to the following manual:

JP1/Network Node Manager i Setup Guide (3021-3-E02-10(E))

2. See 24.1.1 Upgrading NNMi management servers from version 12-10 to upgrade an NNMi management server running version 12-10.

^{24.} Upgrading from NNMi Version 9, 10, or 11

24.2 Upgrading to a different NNMi management server

This section describes the process for upgrading to NNMi 12-50 on a new system while maintaining the configuration of the existing (source) NNMi management server.

Note

Before upgrading an NNMi management server, see 1. Preinstallation Checklists.

The steps below explain how to copy data from the source NNMi management server to a target NNMi management server. These steps assume you have NNMi 12-10 running on the source NNMi management server.

- 1. Back up the source NNMi management server by running the nnmbackup.ovpl script. Label the backup file. You would use this backup only in the unlikely event of a failed migration. For details, see the *nnmbackup.ovpl Reference Page*.
- 2. By following the procedure described in 2. Installing and Uninstalling NNMi, install NNMi 12-50 on the source NNMi management server.
- 3. Verify that NNMi 12-50 is running correctly on the source NNMi management server.
- 4. Back up NNMi 12-50 on the source NNMi management server by running the nnmbackup.ovpl script. Label this backup file.

You will need this file for copying data to the target NNMi management server. For details, see the *nnmbackup.ovpl Reference Page*.

5. By following the procedure described in 2. Installing and Uninstalling NNMi, install NNMi 12-50 on the target NNMi management server.

To migrate the data from step 4, the target NNMi management server must be running the same operating system version. NNMi does not support data migration to an NNMi management server running on a different operating system.

- 6. Run the nnmrestore.ovpl script to copy NNMi database information to the target server. For details, see the *nnmrestore.ovpl Reference Page*.
- 7. Obtain and install a new license on the target NNMi management server.
- 8. Verify that the information from the target NNMi management server migrated successfully from the existing NNMi management server.

24.3.1 NNMi versions supported by global network management

A regional manager running NNMi 12-10 or earlier that is connected to a global manager running NNMi 12-50 is not supported. The global manager and regional managers must be running the same NNMi version.

24.3.2 Global network management upgrade steps

During upgrading of NNMi management servers configured in a global network management environment to NNMi 12-50, the connections between the global network manager and the regional managers will be dropped until both the global managers and the regional managers have been upgraded to NNMi 12-50. For this reason, we recommend that you upgrade all of the servers at approximately the same time to minimize the total downtime.

For example, you might upgrade the NNMi management servers using the following steps:

1. Upgrade the regional managers to NNMi 12-50 and verify that they are operating correctly.

If the certificate repository is not already migrated to the PKCS#12 repository, after upgrade installation, please migrate to the PKCS #12 repository on the regional managers by using the steps in 10.2 Configuring an Upgraded NNMi Environment to Use the New Keystore.

The global manager will remain disconnected while the process of upgrading the regional managers is underway.

2. Upgrade the global manager to NNMi 12-50.

If the certificate repository is not already migrated to the PKCS#12 repository, after upgrade installation, please migrate to the PKCS #12 repository on the global manager by using the steps in 10.2 Configuring an Upgraded NNMi Environment to Use the New Keystore.

Important

Note the following:

• After upgrading has been completed, the updating of statuses and incidents might be delayed.

24.4.1 Upgrading from NNMi 12-10 configured for application failover

To upgrade from NNMi 12-10 while running in an NNMi application failover configuration, follow the steps provided below.

(1) Upgrading to NNMi 12-50 configured for application failover

To upgrade NNMi management servers configured for application failover, follow these steps:

1. As a precaution, run the nnmconfigexport.ovpl script on both the active and the standby NNMi management servers before proceeding.

For details, see 4.2 Best practice: Save the existing configuration.

2. As a precaution, back up your NNMi data on both the active and standby NNMi management servers before proceeding.

For details, see 20.2.2 Backup scope.

3. Complete the steps provided below on the active NNMi management server.

Note that NNMi must be running for the following nnmcluster steps to work. Completing these steps will reduce the time needed for startup of the standby NNMi management server shown in step 7.

a. Run the nnmcluster command.

b. When NNMi prompts you, type dbsync, then press **Enter**. Review the displayed information to make sure it includes the following messages:

- ACTIVE_DB_BACKUP: Reports that the active NNMi management server is performing a new backup.
- ACTIVE_NNM_RUNNING: Reports that the active NNMi management server has completed the backup reported in the previous message.
- STANDBY_RECV_DBZIP: Reports that the standby NNMi management server is receiving a new backup from the active NNMi management server.
- STANDBY_READY: Reports that the standby NNMi management server is ready to perform in the event of a failure of the active NNMi management server.
- c. Run exit or quit to stop the interactive nnmcluster process you started in step a.
- 4. Run the nnmcluster -shutdown command on the standby NNMi management server.

This stops all nnmcluster processes on the standby NNMi management server.

5. To verify there are no nnmcluster nodes running on the standby NNMi management server, complete the following steps on the standby NNMi management server:

a. Run the nnmcluster command.

b. Verify that there are no local nnmcluster nodes present except the one marked (SELF). There might be one or more remote nodes present.

c. Run exit or quit to stop the interactive nnmcluster process you started in step a.

- 6. Complete the following steps on the standby NNMi management server to temporarily disable application failover: **a.** Edit the following file:
 - Windows: %NNM_SHARED_CONF%\props\nms-cluster.properties
 - Linux: \$NNM_SHARED_CONF/props/nms-cluster.properties

^{24.} Upgrading from NNMi Version 9, 10, or 11

- **b.** Comment out the com.hp.ov.nms.cluster.name parameter.
- **c.** Save your change.
- d. Edit the following file:
- Windows: %NNM_DB%\Postgres\postgresql.conf
- Linux: \$NNM DB/Postgres/postgresql.conf

e. Delete the following application failover settings:

- Line beginning with # The following lines were added by the NNM cluster.
- Line beginning with archive command =
- Line beginning with archive_timeout =
- Line beginning with max wal senders =
- Line beginning with archive mode =
- Line beginning with wal level =
- Line beginning with hot_standby =
- Line beginning with wal keep segments =
- Line beginning with listen addresses =

f. Save your changes.

g. Create the following null file:

- Windows: %NNM_TMP%\postgresTriggerFile
- Linux: \$NNM TMP/postgresTriggerFile
- 7. Start and then stop processes on the standby NNMi management server.

a. Run the ovstart command on the standby NNMi management server. Running the ovstart command causes the standby NNMi management server to import the transaction logs from the active NNMi management server.

b. When the ovstart command has finished, run the ovstatus -v command. All NNMi services normally show the state RUNNING.

c. Run the ovstop command on the standby NNMi management server.

8. Upgrade the standby NNMi management server to NNMi 12-50 by following the instructions in 2. Installing and Uninstalling NNMi and the *Release Notes*.

If the certificate repository is not already migrated to the PKCS#12 repository, after upgrade installation, please migrate to the PKCS #12 repository on the former standby NNMi management server by using the steps in 10.2 Configuring an Upgraded NNMi Environment to Use the New Keystore.

You now have the former active NNMi management server running NNMi 12-10 or earlier and the former standby NNMi management server running NNMi 12-50. These NNMi management servers are now running independently with no database synchronization. This means that both NNMi management servers are monitoring the network in parallel.

To complete the upgrade and remedy this situation, select a time to upgrade the former active node to NNMi 12-50. Have the operators temporarily use the former standby node to monitor the network while you complete the upgrade.

The remainder of this procedure assumes you plan to retain the database information from the former active node and discard the database information from the former standby node.

- 9. Run the nnmcluster -halt command on the former active NNMi management server.
- 10. To verify that there are no nnmcluster nodes running on the former active NNMi management server, complete the following steps on the former active NNMi management server:
 - a. Run the nnmcluster command.

^{24.} Upgrading from NNMi Version 9, 10, or 11

b. Verify that there are no local nnmcluster nodes present except the one marked (SELF). There might be one or more remote nodes present.

c. Run exit or quit to stop the interactive nnmcluster process you started in step a.

- 11. Complete the following steps on the former active NNMi management server to disable application failover temporarily:
 - **a.** Edit the following file:
 - Windows: %NNM_SHARED_CONF%\props\nms-cluster.properties
 - Linux: \$NNM_SHARED_CONF/props/nms-cluster.properties
 - **b.** Comment out the com.hp.ov.nms.cluster.name parameter.

c. Save your changes.

12. Upgrade the former active NNMi management server to NNMi 12-50 by following the instructions in 2. Installing and Uninstalling NNMi.

If the certificate repository is not already migrated to the PKCS#12 repository, after upgrade installation, please migrate to the PKCS #12 repository on the former active NNMi management server by using the steps in 10.2 Configuring an Upgraded NNMi Environment to Use the New Keystore.

Now you have two servers running NNMi 12-50, but they are still independent because the databases are not synchronized.

- 13. Complete the following steps on the former active NNMi management server:
 - **a.** Run the ovstop command.
 - **b.** Edit the following file:
 - Windows: %NNM_SHARED_CONF%\props\nms-cluster.properties
 - Linux: \$NNM_SHARED_CONF/props/nms-cluster.properties

c. If you have upgraded from version 11-50, type in the value of the com.hp.ov.nms.cluster.name parameter.

If you have upgraded from version 11-50, the commented-out properties are not preserved, so you must retype the cluster name.

d. Uncomment the com.hp.ov.nms.cluster.name parameter.

e. Save your changes.

- 14. Run either the ovstart command or the nnmcluster -daemon command on the former active NNMi management server. It is now the active cluster node.
- 15. Instruct the operators to begin using the active cluster node to monitor the network.

The former standby NNMi management server discards all database activity that occurred during the maintenance process, from step 9 through step 13.

16. Complete the following steps on the former standby NNMi management server:

a. Run the ovstop command.

b. Edit the following file:

Windows: %NNM_SHARED_CONF%\props\nms-cluster.properties

Linux: \$NNM_SHARED_CONF/props/nms-cluster.properties

c. If you have upgraded from version 11-50, type in the value of the com.hp.ov.nms.cluster.name parameter.

- d. Comment out the com.hp.ov.nms.cluster.name parameter.
- e. Save your changes.

^{24.} Upgrading from NNMi Version 9, 10, or 11

17. Run either the ovstart command or the nnmcluster -daemon command on the former standby NNMi management server.

This NNMi management server becomes the standby node and receives a copy of the database from the active cluster node.

(2) Applying patches to NNMi 12-50 configured for application failover

Both NNMi management servers must be running the same NNMi version and patch level. To apply patches to the active and standby NNMi management servers, use either of the following procedures:

- Apply patches for application failover, shutting down both the active and standby servers Use this procedure when there is no problem with interrupting network monitoring.
- Apply patches for application failover, keeping one NNMi management server active Use this procedure when you must not interrupt network monitoring.

(a) Applying patches for application failover, shutting down both the active and standby servers

This procedure results in both NNMi management servers being inactive for some period of time during the patch installation process. To apply patches on the NNMi management servers configured for application failover, follow these steps:

1. As a precaution, run the nnmconfigexport.ovpl script on both the active and the standby NNMi management servers before proceeding.

For details, see 4.2 Best practice: Save the existing configuration.

2. As a precaution, back up your NNMi data on both the active and the standby NNMi management servers before proceeding.

For details, see 20.2.2 Backup scope.

3. As a precaution, run the following procedure on the active NNMi management server:

a. Run the nnmcluster command.

b. When NNMi prompts you, type dbsync, then press **Enter**. Review the displayed information to make sure it includes the following messages:

- ACTIVE_DB_BACKUP: Reports that the active NNMi management server is performing a new backup.
- ACTIVE_NNM_RUNNING: Reports that the active NNMi management server has completed the backup reported in the previous message.
- STANDBY_READY: Reports the previous status of the standby NNMi management server.
- STANDBY_RECV_DBZIP: Reports that the standby NNMi management server is receiving a new backup from the active NNMi management server.
- STANDBY_READY: Reports that the standby NNMi management server is ready to perform in the event of a failure of the active NNMi management server.
- c. Run exit or quit to stop the interactive nnmcluster process you started in step a.
- 4. Run the nnmcluster -halt command on the active NNMi management server.

This stops all nnmcluster processes on both the active and standby NNMi management servers.

- 5. To verify there are no nnmcluster nodes running on both servers, complete the following steps on both the active and the standby NNMi management servers:
 - a. Run the nnmcluster command.

^{24.} Upgrading from NNMi Version 9, 10, or 11

- **b.** Verify that there are no nnmcluster nodes present except the one marked (SELF).
- **c.** Run exit or quit to stop the interactive nnmcluster process you started in step a.
- 6. On the active NNMi management server, comment out the com.hp.ov.nms.cluster.name parameter in the nms-cluster.properties file:
 - a. Edit the following file:
 - Windows: %NNM_SHARED_CONF%\props\nms-cluster.properties
 - Linux: \$NNM_SHARED_CONF/props/nms-cluster.properties
 - **b.** Comment out the com.hp.ov.nms.cluster.name parameter.
 - c. Save your change.
- 7. Apply the NNMi patches on the active NNMi management server using the instructions in RELEASE. TXT provided with the patches.
- 8. On the active NNMi management server, uncomment the com.hp.ov.nms.cluster.name parameter in the nms-cluster.properties file:
 - a. Edit the following file:
 - Windows: %NNM_SHARED_CONF%\props\nms-cluster.properties
 - Linux: \$NNM_SHARED_CONF/props/nms-cluster.properties
 - **b.** Uncomment the com.hp.ov.nms.cluster.name parameter.
 - c. Save your change.
- 9. Run the ovstart command on the active NNMi management server.
- 10. Verify that the patches have been installed correctly on the active NNMi management server by viewing information on the **Product** tab of the **Help** > **System Information** window in the NNMi console.
- 11. Run the nnmcluster -dbsync command to make a new backup.
- 12. On the standby NNMi management server, comment out the com.hp.ov.nms.cluster.name parameter in the nms-cluster.properties file, as shown in a through c in step 6.
- 13. Apply the NNMi patches on the standby NNMi management server.
- 14. On the standby NNMi management server, uncomment the com.hp.ov.nms.cluster.name parameter in the nms-cluster.properties file, as shown in a through c in step 8.
- 15. Run the ovstart command on the standby NNMi management server.

(b) Applying patches for application failover, keeping one NNMi management server active

This procedure results in one NNMi management server always being active during the patch installation process.

Although this method preserves continuous monitoring of the network, NNMi loses the transaction logs that are generated during this patch installation process.

To apply NNMi patches on the NNMi management servers configured for application failover, follow these steps:

1. As a precaution, run the nnmconfigexport.ovpl script on both the active and the standby NNMi management servers before proceeding.

For details, see 4.2 Best practice: Save the existing configuration.

2. As a precaution, back up your NNMi data on both the active and the standby NNMi management servers before proceeding.

^{24.} Upgrading from NNMi Version 9, 10, or 11

For details, see 20.2.2 Backup scope.

- 3. Run the nnmcluster command on one of the nodes.
- 4. Enter dbsync on the NNMi management server used in the previous step to synchronize the two databases. The dbsync option works on an NNMi management server using an embedded database.
- 5. Wait until the active NNMi management server reverts to ACTIVE_NNM_RUNNING and the standby NNMi management server reverts to STANDBY_READY before continuing.
- 6. Exit from or quit the nnmcluster command.
- 7. Stop the cluster on the standby NNMi management server by running the following command on the standby NNMi management server:

nnmcluster -shutdown

8. Make sure the following processes and services have terminated before continuing:

```
- postgres
```

- ovjboss
- 9. Make sure the nnmcluster process has terminated before continuing.

If the nnmcluster process will not terminate, manually kill the nnmcluster process only as a last resort.

- 10. Edit the following file on the standby NNMi management server:
 - Windows: %nnmDataDir%shared\nnm\conf\props\nms-cluster.properties
 - Linux: \$nnmDataDir/shared/nnm/conf/props/nms-cluster.properties
- 11. Comment out the cluster name property by placing a hash mark (#) at the beginning of the line, then save your change:

#com.hp.ov.nms.cluster.name = NNMicluster

- 12. Apply the NNMi patches on the standby NNMi management server.
- 13. At this point, the standby NNMi management server has been patched but is stopped, and the active NNMi management server has not been patched and is running.

Stop the active NNMi management server and immediately bring the standby NNMi management server online to monitor your network.

14. Shut down the cluster on the active NNMi management server by running the following command on the active NNMi management server:

nnmcluster -halt

15. Make sure the nnmcluster process has terminated.

If it does not terminate within a few minutes, manually kill the nnmcluster process.

- 16. On the standby NNMi management server, uncomment the cluster name from the nms-cluster.properties file.
- 17. Start the cluster on the standby NNMi management server by running the following command on the standby NNMi management server:

nnmcluster -daemon

- 18. Install the NNMi patches on the active NNMi management server.
- At this point, the previous active NNMi management server has been patched but is offline.
 Bring it back into the cluster (as the standby NNMi management server) by performing the following:

^{24.} Upgrading from NNMi Version 9, 10, or 11
- a. Uncomment the entry in the nms-cluster.properties file on the active NNMi management server.
- **b.** Start the active NNMi management server with the following command:

nnmcluster -daemon

20. To monitor the progress, run the following command on both the active and the standby NNMi management servers:

nnmcluster

Wait until the previous active NNMi management server finishes retrieving the database from the previous standby NNMi management server.

21. After the previous active NNMi management server displays STANDBY_READY, run the following command on the previous active NNMi management server:

nnmcluster -acquire

^{24.} Upgrading from NNMi Version 9, 10, or 11



Comparison with NNM Version 8 or Earlier

This chapter explains the important differences between NNMi and NNM version 8 or earlier. If you have been using an earlier version, read this chapter to plan and configure NNMi. If you are using NNMi for the first time, you do not need to read this chapter.

25.1 Network discovery

NNMi collects information about the elements in your network (such as devices, nodes, and their components) that are added to the database. In NNMi, *inventory discovery* means searching for new nodes, while *Layer 2 discovery* means connectivity modeling.

Upon startup, NNM in its default configuration used its own loopback address as a seed to start automatic discovery of the directly connected network (based on its own IP address and subnet mask). In contrast, NNMi allows the administrator to exert control from the start. For NNMi auto-discovery, you define the discovery area based on an IP address range before you perform discovery, and you specify at least one seed device (usually a router).

The following figure shows in the center the tools, files, and commands used by NNMi for configuring discovery, surrounded by the NNM tools, files, and commands.





25.1.1 Key concepts of discovery

This subsection describes briefly the major changes that were made from NNM to NNMi. For details about discovery in NNMi, see *Discovering Your Network* in NNMi Help.

- All information is stored in a single relational database.
- An easily configurable integrated discovery engine is used.

25. Comparison with NNM Version 8 or Earlier

- The spiral discovery process enables you to continuously update the topology information as changes occur in the network. This enables you to discover topology changes (both inventory and Layer 2) more frequently than would be possible using a set re-discovery interval.
- All discovery-target nodes are counted toward the node count restriction set by the license, regardless of the management mode (MANAGED, UNMANAGED, or OUT OF SERVICE). The number of nodes that are discovered cannot exceed the license restriction.
- Auto-discovery has the same meaning in both NNMi and NNM, but the configuration approach differs.
 - In NNMi, you must first define an auto-discovery boundary, specify at least one IP address, and then run discovery.

- NNMi auto-discovery uses an expanding model. NNMi auto-discovery discovers and manages all routers, switches, and subnets within the specified boundary. In NNMi, you specify the additional device types to be discovered and managed.

Note

In the default, only SNMP nodes are discovered.

- Seeded discovery has the same meaning in both NNMi and NNM, but the configuration approach differs.
 - In NNMi, you specify a discovery seed through the user interface.
 - NNM seed files can be used in NNMi without any modifications.
 - The nnmloadseeds.ovpl command in NNMi replaces the loadhosts command in NNM.
- NNMi configuration polling (nnmconfigpoll.ovpl) replaces NNM demand polling (nmdemandpoll) used for determining the device configuration information.

^{25.} Comparison with NNM Version 8 or Earlier

25.2 Status monitoring

Status monitoring can visualize the latest network in relation to devices and components that could fail. If polling of a configuration element fails, NNMi investigates the cause and sends out a root cause alarm to the incident browser.

The following figure shows in the center the tools, files, and commands used by NNMi for configuring status monitoring, surrounded by the NNM tools, files, and commands.





25.2.1 Key concepts of status monitoring

This subsection describes briefly the major changes that were made from NNM to NNMi. For details about NNMi status monitoring, see *Monitoring Network Health* in NNMi Help.

- You complete the configuration through the user interface.
- NNMi node groups and interface groups replace the topology filter.
 - You can filter groups using only predefined attributes.
 - You cannot link groups using Boolean operators.
 - Node groups use a device filter instead of the sysObjectId wild card.
 - You can restrict the interface groups based on the node groups and interface types to be hosted.
- A wide range of control functions makes it easy to exclude unneeded interfaces.

^{25.} Comparison with NNM Version 8 or Earlier

- NNMi matches the monitoring configuration according to the level of detail, from highly specific to more general, in order, such as (first) interface configuration, (second) node configuration, and (third) default configuration.
- To change the monitoring action throughout the entire system, change all configurations at all levels.
- NNMi's status polling (by selecting Actions > Polling, and then Status Poll or by running the nnmstatuspoll.ovpl command) replaces NNM's demand polling (by running the nmdemandpoll command) for determining device status.
- As the default, NNMi polls only interfaces that are connected to another known interface through Layer 2 connectivity. You can enable polling of unconnected interfaces and interfaces that are hosting IP addresses.

^{25.} Comparison with NNM Version 8 or Earlier

NNMi has a central location called the incident view, where you can view managed events and SNMP traps. You can control the SNMP traps that are displayed as incidents.

The following figure shows in the center the tools, files, and commands used by NNMi for configuring event monitoring, surrounded by the NNM tools, files, and commands.





25.3.1 Key concepts of event monitoring

This subsection describes briefly the major changes that were made from NNM to NNMi. For details about NNMi incidents, see *Configuring Incidents* in NNMi Help.

- Event subsystems are not used in process-to-process communication. For this reason, the event volume has been reduced significantly. The administrator no longer needs to specify whether to display or log individual IPC messages.
- Only traps configured to be received are received. Traps that are not configured are filtered out of the event pipeline.
- All received traps are displayed.
- The trap filter of the NNMi event subsystem process is configured based on what is selected on the **Incident Configuration** form.
- NNMi's nnmincidentcfg.ovpl command loads only the trap definition of the specified MIB module.

25. Comparison with NNM Version 8 or Earlier

- Pairwise correlation, rate correlation, and deduplication correlation that occur in the event pipeline are provided (NNMi does not include an event correlation system (ECS)).
- You can configure the actions that will occur in an incident's lifecycle. You can configure all scripts, executable files, or Jython actions as actions.

^{25.} Comparison with NNM Version 8 or Earlier



Upgrading from NNM Version 8 or Earlier

This chapter explains the basic method of upgrading from NNM to NNMi. This method is useful for many users. This chapter, however, does not explain advanced migration topics or customization.

26.1.1 New NNM system

NNM has been used in a wide variety of network environments with several generations of software. Users who have been using NNM version 5 or earlier in the router-centric world must be facing a large burden that does not match the current network structure. If your NNM system is at least two years old, we recommend that you take this opportunity to start a new system. By reassessing how best to manage the current network, you might be able to achieve significant overhead reduction and operational efficiency improvement compared to NNM.

To newly install and begin using NNMi, install NNMi by following the instructions described in 2. Installing and Uninstalling NNMi. Next, study the installation tasks explained in other chapters of this manual. In this case, you need not read this chapter.

26.1.2 Migrating in phases

For some organizations, their system might function better if phased migration is used rather than building an entirely new system. For such organizations, it will be necessary to completely replicate the existing NNM system to a new NNMi system, thus replacing the old system. Although many migration methods are available, we recommend the following phases:

- 26.2 Phase 1: Migrating the SNMP information Configure NNMi using the SNMP information of the environment currently being used.
- 26.3 Phase 2: Migrating discovery

Configure NNMi to discover objects in a manner similar to the way discovery worked under NNM (automatically).

- 26.4 Phase 3: Migrating status monitoring Configure a state polling interval and protocol most appropriate to the environment being used.
- 26.5 Phase 4: Migrating event configuration and event reduction

Configure NNMi to display event severity, category, and messages and to run automatic actions as NNM was configured to. You might also need to configure deduplication, rate counting, and pairwise cancellation.

• Phase 5: Migrating graphical visualization Migrate the hierarchical structures of the NNM location map, Internet submap, and segment submap as NNMi node group configurations. For details about the migration method, see the *Release Notes*.

Table 26-1: Migration range provides an overview of the simplest method and the most detailed and thorough method with respect to the migration range.

- With the simplest method, you import the information unique to your NNM environment and use the NNMi default values for other configurations, which have been improved from NNM.
- With the most detailed and thorough method, carefully check the NNM configurations and replicate them to NNMi.

The remainder of this chapter explains one by one the processes of migrating an NNM configuration to NNMi. The following headings, such as *Collect from NNM* and *Replicate to NNMi*, indicate where in the migration process that portion of the procedure is applied.

- Collect from NNM indicates a task that is performed at the NNM management station.
- Replicate to NNMi indicates a task that is performed at the NNMi management station.

• *NNMi enhancement* indicates an additional task that is performed at the NNMi management station. You can make enhancements at any time, either during or after migration.

At some points, multiple methods are provided, depending on the degree of task difficulty.

Phase	Simplest method	Most detailed and thorough method
SNMP information	 Export all community strings currently being used. Import these community strings into NNMi. NNMi determines the node each community string matches. 	 Export all community strings currently being used. Correct the exported data file and import it into NNMi as the community strings of specific nodes.
Discovery	 Export a list of all discovered nodes. Modify the data file and import into NNMi the file content as seeds without any automatic discovery rule. 	 Identify the methods (seeds, load hosts, filters, and other tools) NNM and netmon used to discover nodes. Use the seeds and the auto-discovery tool to replicate these methods on NNMi as closely as possible.
Status monitoring	The NNMi default values are improved to suit most user requirements. Because there is no need to significantly change these default values, begin operation using the improved default values.	 For each group of nodes, carefully examine which polling interval and polling policy are being used by NNM and netmon. Create NNMi node groups and interface groups so as to replicate the polling interval and polling policy.
Event configuration and event reduction	 Start with the NNMi default configuration. Add custom trap definitions for managed devices. If necessary, add automatic processing. 	 For each type of trap and event, examine what NNM customization was performed. On the NNMi system, customize the matching traps and event types.

Table 26-1: Migration range

26. Upgrading from NNM Version 8 or Earlier

26.2 Phase 1: Migrating the SNMP information

To establish communication with the managed devices, migrate the SNMP community string information used by NNMi.

If the NNM configuration contains IP addresses or host names to be excluded from name resolution, copy that information to NNMi.

Customize the NNMi device profiles for the custom devices of the network being used.

26.2.1 Configuring SNMP access

To collect individual information related to the configuration and connectivity of managed nodes, NNMi discovery needs SNMP access to those nodes. SNMP is used also during status monitoring in order to access the operating states of nodes and the objects contained in them.

Note

NNM tries the community strings one by one in the order they are listed in the configuration of the matching area, and uses the first community string verified to be available. In contrast, NNMi tries all configured community strings in parallel, and uses the first community string verified to be available. If multiple values are available, configure the most suitable community string.

Collect from NNM

The NNM management station has the configuration information needed by SNMP to access the devices in the environment being used.

1. To export the NNM SNMP configuration, perform either of the following operations:

- Open the user interface, choose **Options**, **Configure SNMP**, and then click **Export**. For the target file name, specify snmpout.txt.
- Run the following command: xnmsnmpconf -export > snmpout.txt

Example of NNM SNMP information

An output example follows:

```
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 has the following fields separated by a colon (:):

```
target:community:proxy<sup>#1</sup>:timeout<sup>#2</sup>:retries:poll interval<sup>#3</sup>:port:set-
community:
```

- #1: An asterisk (*) indicates that the item is not a proxy.
- #2: Expressed in 1/10 seconds.

#3: Expressed in seconds.

To see the detailed information for a value, use the following command (do not use it for import, however): xnmsnmpconf -export -verbose

For details about the ovsnmp.conf file format, see the *ovsnmp.conf Reference Page*.

- 2. Check the configured community string in the following file:
 - Windows: %OV_CONF%\netmon.cmstr
 - Linux: \$OV_CONF/netmon.cmstr

Replicate to NNMi

Select methods for entering community strings in NNMi. Use these methods starting with the unique communication string list in the snmpout.txt file created in step 1 under *Collect from NNM*.

Note

You cannot migrate the settings in the SNMP Proximity System and Configuration Community Name configuration areas.

(1) Simple method

The simplest method is to enter all NNM community strings so that NNMi can resolve the SNMP community strings used by individual devices. Community string discovery is enabled by default. This function enables quick migration.

- 1. Inform the network operating center (NOC) to expect authentication errors during the initial discovery by NNMi. NOC personnel can ignore these authentication errors during this period.
- 2. Perform one of the following operations:
 - Modify snmpout.txt to match the format used by NNMi. Next, use NNMi to load these values.
 - Using the snmpout.txt file as a sample, manually build an NNMi input file. Next, use NNMi to load these values.
 - Enter the values into the NNMi console according to the following procedure:

a. Examine the unique community string value list in the snmpout.txt file:

- Windows: Open the snmpout.txt file using Excel. First select data rows and then sort them by column B.

In this example, the following two unique community strings are considered:

public

mycommstr

- Linux: Run the following command:

cut -f 2 -d ':' < snmpout.txt | sort -u</pre>

b. In the NNMi console, from the **Configuration** workspace, choose **Communication Configuration**. Enter all unique values in the **Default SNMPv1/v2 Community Strings** tab.

c. Set up the timeout, retry count, and ports.

Communication Configuration * 🗴						
Default SNMP Settings For more information, click here. Enable SNMP Address Rediscovery Get-Bulk Enabled	^	Default SNMPv1/v2 Community Strings Read Community Strings For more information, click here. * *	Default SNMPv3 Settings	Default Device Cred ← 1-2 of 2 →	→ H	~
* SNMP Timeout 5.00 Seconds 💌		▲Read Community String				Orderir
Example: Timeout = 3 seconds, Retries Count = 4. NNMi attempts to communicate using SNMP and waits 3 seconds for an answer. Each additional attempt, NNMi adds 3 seconds before trying again, trying at 6, 9, and 12 for a total of 30 seconds. See online Help for more information.		mycommstr public				
* SNMP Retries Count 2						
SNMP Port 161 SNMP Proxy Address SNMP Proxy Port						
SNMP Minimum Security Level Community						

(2) Modified simple method

Group the community strings for each IP area to be used. Enter the values in each area in the NNMi console so that NNMi determines the SNMP community string to be used for each device. There will be fewer authentication failures than the aforementioned simple method.

- 1. Check the unique values list in the snmpout.txt file for each IP used by NNM.
- 2. In the NNMi console, from the **Configuration** workspace, choose **Communication Configuration**. After creating IP areas, enter community strings in each area.
- 3. Set up the timeout value, retry count, and ports.

(3) Automated method

After converting the snmpout.txt file into the format needed by the nnmcommload.ovpl command, load the individual community string being used by each device.

- 1. To make the snmpout.txt file conform for use by NNMi tools, perform either of the following operations:
 - Use an editor to create a file that is appropriate to NNMi. The result follows:

```
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
```

• Run the following command (applicable in Linux only):

awk 'BEGIN {FS = ":" }; {printf"%s, $s \in n$ ", 1, 2 }' \ <snmpout.txt> mysnmp.txt This command is valid only for configuring the individual nodes in the file. Manually delete the range and wildcard settings.

2. Run the following command:

```
nnmcommload.ovpl -u username -p password -file mysnmp.txt
```

3. In the NNMi console, configure the default community string and a community string for the IP range.

4. In the NNMi console, set up the timeout value, retry count, and ports.

(4) Method using NNMi

In the NNMi console, from the Configuration workspace, choose Communication Configuration.

Replicate the configured values of the snmpout.txt file.

NNMi enhancement

Using the following information, enhance NNMi's communication access configuration:

- Host name wildcard (if it is more suitable to the environment than an IP range)
- Global default, IP range, and ICMP timeout and retry count for certain nodes
- Enabling or disabling SNMP or ICMP access to a specific area in the network
- Management addresses given higher priority for certain nodes

Note

NNM selects the smallest loopback address when selecting a management address. NNMi also selects the smallest loopback address.

26.2.2 Restricting name resolution

If you know the restrictions on the DNS (or other name resolution) service, you can instruct NNM and NNMi to avoid looking up these devices. If this step does not apply to your system, go to 26.2.1 Configuring SNMP access.

Collect from NNM

- 1. Check the following file and identify the addresses that NNM excludes from address-to-host-name name resolution.
 - Windows: %OV_CONF%\ipnolookup.conf
 - Linux: \$OV_CONF/ipnolookup.conf
- 2. Run the following command and check for the host names NNM excludes from name-to-address name resolution:

snmpnolookupconf dumpCache > snmpnolookup.out

Replicate to NNMi

- 3. Add the addresses from step 1 to the following file:
 - Windows: %NnmDataDir%shared\nnm\conf\ipnolookup.conf
 - Linux: \$NnmDataDir/shared/nnm/conf/ipnolookup.conf
- 4. Add the host names from step 2 to the following file:
 - Windows: %NnmDataDir%shared\nnm\conf\hostnolookup.conf
 - Linux: \$NnmDataDir/shared/nnm/conf/hostnolookup.conf

For details about the format of these configuration files, see the *ipnolookup.conf* and *hostnolookup.conf Reference Pages*.

NNMi enhancement

NNMi performs lookup only during discovery. Replicating the NNM nolookup configuration to NNMi automatically improves the spiral discovery action.

5. In NNMi, you can select the DNS host name, IP address, or MIB II sysName to be used as the name label to be displayed. Use the following procedure:

a. In the NNMi console, open the Configuration workspace.

- **b.** Choose **Discovery** > **Discovery** Configuration.
- c. In the Node Name Resolution area, set up the node name priority.

26.2.3 Customizing device profiles

Through SNMP communication with devices, NNM directly collects several configuration information items. Additionally, some information is derived from the system object ID (sysObjectID) of devices.

Mapping of sysObjectID to NNMi attributes is done using a device profile. A device profile is used for grouping nodes to be monitored, filtering nodes to be displayed, and grouping nodes into categories for maintenance of discovery.

You cannot migrate the following configuration areas:

- Customized symbols
- · Customized database fields and default values

Note

NNMi provides many device profiles by default and they are available after installation. You can add any device profiles that are determined by using NNMi device discovery to be missing. Migration of device profiles is optional.

If you want to maximize matching of device profiles and NNM settings, use the procedure below to add device profiles to NNMi.

Collect from NNM

- 1. Determine whether the OID file has been customized for the NNM version being used.
 - In NNM 07-10 or earlier, oid_to_sym, id_to_type, and HPoid2type are used to map the system's sysObjectID to a database attribute and the symbol to be displayed.
 - In NNM 08-00 or later, the oid_to_sym file is replaced with the oid_to_sym_registration directory structure.

Replicate to NNMi

Because NNMi provides many pre-configured device profiles for known system object IDs, you can use the necessary device profiles right away. With the simplest method, you start a discovery process, check the results, and make only the necessary changes.

2. In the NNMi console, from the Configuration workspace, choose Device Profiles.

For each customized value, find an entry using sysObjectID.

- 3. If necessary, update the device profiles configuration.
 - For the entries provided by NNMi, verify that the values that have been set match the NNM attributes.

- For the entries not provided by NNMi, create a new device profile for sysObjectID.
- 4. After the first discovery, sort the **Device Profile** column in the node inventory and find nodes with No Device Profile.

The profile type called No Device Profile indicates that sysObjectID has not yet been configured in NNMi. NNMi applies the default-monitoring configuration to nodes with No Device Profile. Note that it is difficult to filter these nodes.

You can build new device profiles so that device profiles are defined for all sysObjectIDs in the NNMi database.

^{26.} Upgrading from NNM Version 8 or Earlier

26.3 Phase 2: Migrating discovery

Migrate the discovery schedule and configuration. NNMi spiral discovery starts immediately when one or more discovered seeds are saved.



) Important

Start discovery after configuring NNMi to use the appropriate community strings for the network environment.

After the first discovery by NNMi is finished, migrate the device-to-device connections that were configured manually in NNM.

26.3.1 Configuring a discovery schedule

NNM discovery processes can be executed independently. To migrate discovery to NNMi, all you need to do is to transfer the interval at which NNM discovers nodes.

The following schedule configuration areas are not used in NNMi and cannot be migrated:

- Connector device topology check. Topology checking is now performed automatically every time NNMi finds a trigger indicating the possibility of a change.
- Configuration check. NNMi now performs a configuration check at the time of scheduled discovery or based on various triggers.
- Layer 2 (expanded topology) discovery action. Because NNMi performs Layer 2 discovery when it finds a device, there is no need to schedule this action separately.
- Automatic adjustment of the discovery polling interval

Collect from NNM

1. Determine when NNM is performing rediscovery.

a. In the user interface, choose **Options** and then **Network Polling Configuration**.

b. On the IP Discovery tab, check Discovery Polling Interval.

- If a fixed interval is used, make a note of the value so you can set it in NNMi.

- If NNM is using an automatically adjusted interval, NNM stands by for a maximum of 24 hours. You can leave NNMi at the default value of 24 hours or you can select a new value.

- If auto-discovery is not enabled, check the cycle in **Perform Configuration Checks** on the **General** tab, and make a note of the value so you can set it in NNMi.

Configuration Area: General 🖃
General Discovery Options Use Discovery Filter I Biscover Beyond License Limit
Topology Polling Interval
Configuration Checks
OK Roply Defaults Cancel Help

Replicate to NNMi

2. In the NNMi console, from the **Configuration** workspace, choose **Discovery** > **Discovery Configuration**, and then set in **Rediscovery Interval** the value determined in step 1.

Discovery Configurat	tion x											
 Global Control 		<	Schedule Settings	Auto-Discovery Rules	Subnet Connection Rules	Unnumbered Ir , 🗸						
Enable Discovery of						A						
ATM/Frame Relay Interfaces for		Default Interval Setting										
Monitoring		* Rediscovery Interval 1 Days -										
Auto-Discovery Ping	g Sweep Control (IPv4 only)	No	ode Group Interval Se	ttings								
This control can overrid Discovery Rules.	de the Enable Ping Sweep choice for all Auto-	N	ode Group			- 🗊 -						
Ping Sweep	None -	Re	ediscovery Interval	24.00 Hours *								
Sweep Interval	24.00 Hours -	De	elete Unresponsive Ol	bjects Control								
Node Name Resolut	ion	do l	Mi deletes objects from NOT delete unresponsi	the NNMi database after the objects. For more inform	he specified number of unresp tation, click here.	onsive days. Zero (0) means						
First Choice	Short DNS Name 👻	Pe	eriod (in Days) to	-	-							
* Second Choice	Short sysName 💌	• De No	elete Unresponsive odes	0								
Third Choice	IP Address 👻	Pe	eriod (in Days) to	0								
Layer 2 Connection Source			at are Down	0								
Node Group to disabl FDB	le 🖉 🗸											
NNMi enhanc	ement											

All other configuration updates are added automatically, making configuration simpler and discovery more efficient than in NNM.

26.3.2 Selecting a discovery method

Determine which of the following models to use for NNMi discovery:

• Seeded discovery without auto-discovery rules. In this type of discovery, because the administrator adds only the necessary nodes to the seeds, the nodes to be discovered can be restricted. Perform only the operation described in the following subsection:

- 26.3.4 Adding seeded discovery

- Perform the operations described in both of the following subsections:
 - 26.3.3 Setting up auto-discovery rules
 - 26.3.4 Adding seeded discovery

For details about the differences between the NNMi discovery methods, see *Determine Your Approach to Discovery* in NNMi Help.

Note

NNM licenses are counted based on the number of nodes being managed (nodes whose status is monitored). In contrast, NNMi licenses are counted based on the number of nodes deployed in the discovered topology (both monitored and non-monitored nodes).

Because of this difference, some users might wish reduce the number of discovered nodes. However, also including non-monitored nodes in the database offers some benefits.

Examples:

- Even when you are not responsible for managing devices, you can display the service provider's access routers and connections to them.
- The status-monitoring algorithm is based on the connections displayed in the database. An interface in which the device at the other end of the link is not in the database is not monitored. You can change the default status-monitoring configuration or discover that device. Which one to select depends on the part of the usage environment in which you are interested. For details, see 7.2.4 Interfaces to Unmonitored Nodes.

26.3.3 Setting up auto-discovery rules

Configuring NNMi discovery provides an excellent opportunity for thinking about the NNMi management targets. Before configuring NNMi discovery and converting filters, assess the current network environment and consider what to incorporated into the NNMi topology.

For direct conversion, the NNMi discovery rules include the following two sets of tasks: expanding the discovery scope, and limiting the objects discovered within the scope.

^{26.} Upgrading from NNM Version 8 or Earlier

Note

When configuring NNMi, it is important to first define all rules that expand or restrict discovery and then to enter the seed for starting the discovery process.

Because the following schedule configuration areas are not used in NNMi, they cannot be migrated:

- IPX discovery from Windows
- Discoveries that exceed the license restrictions
- Disabling Layer 2 object discovery (this is always enabled in NNMi)
- Exclusion of discovery using an attribute filter other than IP address and sysObjectID (and its derivatives)
- Restriction on Layer 2 discovery based on the CDP protocol area (for example, integrated port and vlan)
- Expanded topology configuration (no longer corresponds to NNMi's spiral discovery).

(1) Configuring spiral discovery

In NNMi, two methods are available for configuring spiral discovery. One is manual loading of nodes (from a host file, for example) and the other is using auto-discovery rules.

(a) Manual loading of nodes

```
Collect from NNM
```

1. In NNM, look for a file that includes the content entered in the loadhosts command.

This file lists the IP addresses and host names of the individual nodes, as well as their subnet masks if specified.

Example of NNM loadhosts

An example of a loadhosts command file follows:

```
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 way the loadhosts command was used in NNM. To do this, specify the -f option and a seed file, and run the nnmloadseeds.ovpl command.

Note

- Before configuring seeding in NNMi, complete the configuration of all community strings.
- To obtain the same discovery results as NNM's loadhosts command, disable the auto-discovery rules that are set in NNMi. To disable the auto-discovery rules, do one of the following:
 - Delete the rules from the **Discovery Configuration** form.
 - In the **Auto-Discovery Rule** form, clear the **Discover Matching Nodes** check box. The NNMi seed file format has an IP address or a node name (with a comment if provided) for each row. For details, see the *nnmloadseeds.ovpl Reference Page*.

^{26.} Upgrading from NNM Version 8 or Earlier

NNMi seed file example

An example of an NNMi seed file with the same functions as the NNM loadhosts command and host file follows:

10.2.32.201 # comment
10.2.32.202 # comment
1nt03.mycorp.net # comment
1nt02.mycorp.net
10.2.32.205

Q Тір

In NNMi, a loopback address is always given a higher priority as a management address. If a loopback address is not used, NNMi will probably (but not always) use a seed address as a management address. Therefore, a good approach is to copy the hosts file in which the high-priority IP address is written. If you are using a host name, verify that DNS resolves it as a high-priority address. However, this does not guarantee that NNMi will use this address as a management address. For details about how to select a management address, see *Discovery Node Name Choices* in NNMi Help.

(b) Using auto-discovery rules

Collect from NNM

- 1. Check whether a discovery filter was used in NNM.
 - In NNM, a single discovery filter is applied to the entire discovery scope.
 - **a.** Open the NNM user interface.
 - b. Under Options, choose Network Polling Configuration.

c. On the **General** tab, check the **Use a discovery filter** check box. If it is selected, write down the name of the discovery filter being used. If the check box is cleared, proceed to 26.3.4 Adding seeded discovery.

d. Look for a discovery filter in the following file:

- Windows: %OV_CONF%\C\filters

- Linux: \$OV_CONF/C/filters

Check the logic carefully. In NNMi, you can filter the IP address range and system object ID range. You can also migrate some objects, by converting a host name wildcard to an IP range or converting vendor names to a system object ID range.

NNM discovery filter examples

Shown below are NNM filter examples, such as routers, bridges, Nokia_Firewalls, NetBotz, and NetsNSegs. NetBotz firewall and Nokia firewall are defined with 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" ~ .1.3.6.1.4.1.94.1.21.2.10.11 ) ) ||
(isNode && ( "SNMP sysObjectID" ~ .1.3.6.1.4.1.94.1.21.2.10.12 ) ) ||
(isNode && ( "SNMP sysObjectID" ~ .1.3.6.1.4.1.94.1.21.2.10.12 ) ) ||
(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 to NNMi

2. In the NNMi console, enter discovery filters.

NNMi discovery filter entry example

For example, to move the NNM filters shown in *NNM discovery filter examples* in step 1 under *Using autodiscovery rules*, define the following three auto-discovery rules: One for the Nokia firewall, another for NetBotz devices, and a third for routers and switches (collectively referred to as bridges in NNM 08-00 and later). In NNMi, NetsNSegs is not necessary. In this example, the range of the network to be discovered is assumed to be 10.*.*.*.

a. For the Nokia file, enter the rule name (Nokia_Firewalls) and then enter the network IP range 10.*.*.*.

C Changes are	not committed until the top-level form is saved!	
 Basics Auto-Discovery Rul 	les apply only to Default Tenant.	IP Ranges System Object ID Ranges Auto-Discovery Starting Point for this Rule
Name Ordering Notes	Nokia_Firewalls	(IPv4 only) if enabling Ping Sweep for this rule, do not specify more than a maximum of the last two octets (/1 of the network within one Rule. Click here for more information about Ping Sweep for an Auto-Discovery Rule Use Ping Sweep Instead of or In Addition to Discovery Seeds (IPv4 only)
Purpose of the second sec	his Auto-Discovery Rule	Enable Ping Sweep
If enabled, NNMi di criterion. If disabled criterion. Click here Discover Matching	iscovers any Node that complies with this Rule's d, NNMi rejects any Node that complies with this Rule's e from more information.	Specify the IP Address Ranges for this Rule to include. You can also specify subsets of those IP addresses to this Rule to ignore (remain available for another Rule). Click here for more information. Tip: Provide one seed for each WAN's IP Address Range.
Extend Defa	wit Behavior (beyond Routers and Switches)	★ 〒 ② ↑ 章 H ← 1-1 of 1 → H ▲ IP Range Range Type
If enabled, NNMi di complies with this P Discover Any SNM Device	iscovers any Node that responds to SNMP and Rule's criterion. Click here from more information.	10.*.*.* Include in rule

b. Enter each sysObjectID (without the period at the beginning). Next, select the **Discover Any SNMP Device** check box (by default, NNMi discovers only switches and routers). Because these devices might not be marked as switches or routers in some cases, select the **Discover Any SNMP Device** check box when specifying sysObjectIDs.

Auto-Discovery Rule	*								
	C 🖬								
(i) Changes are not	committed until the top-level form is saved!								
· Basics			IP Ranges	System Object ID I	Ranges				
Auto-Discovery Rules a	apply only to Default Tenant.			1					
* Name	Nokia_Firewalls		The System	Object ID Range filter	is '.1.3.6.1' unle	ss specified otherwi	se.		
 Ordering 	110								
Notes			If you provide one of each t	e one or more System type of range. If you pr	Object ID Rang ovide only one of	es and IP Address I or more System Obj	Ranges, each d ect ID Ranges (scovered Node m no IP Address Rar	ust match liges),
			then 'Discove	er Included Nodes' mu	st = 'Reject Nod	les'.			
			Click here for	r more information abo	ut System Obje	ct ID Ranges.			
 Purpose of this A 	Auto-Discovery Rule								
If enabled, NNMi discov criterion. If disabled, NN	vers any Node that complies with this Rule's VMi rejects any Node that complies with this Rule		A Sustem	Object ID Prefix	Danne Tuni		Notes	0000 7 h	
criterion. Click here from	m more information.		136141	1 94 1 21 2 1 1	Include in ru	le.	Hotes		
Discover Matching	~		1.3.6.1.4.1	94.1.21.2.1.10	Include in ru	le			1000
Nodes			1.3.6.1.4.1	.94.1.21.2.1.138	Include in ru	le			
· Extend Default B	ehavior (beyond Routers and Switches)		1.3.6.1.4.1	.94.1.21.2.1.9	Include in ru	le			
If enabled, NNMi discov	vers any Node that responds to SNMP and		1.3.6.1.4.1	1.94.1.21.2.10.11	Include in ru	le			
complies with this Rule	's criterion. Click here from more information.		1.3.6.1.4.1	1.94.1.21.2.10.12	Include in ru	le			
Discover Any SNMP Device	✓		_						_
If enabled, NNMi discov	vers any Node that responds to ICMP and								
complies with this Rule	's criterion. Click here from more information.								
Discover Non-SNMP Devices									
Rule Criterine		~			Tota	al: 6 Selected: 0	Filter: OFF	Auto refresi	h: OFF
THE STREETST									

c. Enter the NetBotz rule. You must specify an IP range here as well. In this rule, the wildcard is used in NNM .1.3.6.1.4.1.5528.*.. In NNMi, an asterisk (*) is implied and therefore is not necessary.

Auto-Discovery Rule * 3										
	0									
(i) Changes are not co	mmitted until the top-level form is saved!									
· Basics		IP Ranges	System Object ID	Ranges						
Auto-Discovery Rules app	ly only to Default Tenant.	•	1							
* Name	NetBotz	The System	Object ID Range filter	is '.1.3.6.1' u	nless specified oth	erwise.				
Notes	120	If you provide one or more System Object ID Ranges and IP Address Ranges, each discovered Node must ma one of each type of range. If you provide only one or more System Object ID Ranges (no IP Address Ranges), then 'Discover Included Nodes' must = 'Reject Nodes'.								
 Purpose of this Auto 	o-Discovery Rule	Click here for	r more information abo	out System O	bject ID Ranges.					
If enabled, NNMi discover criterion. If disabled, NNM criterion. Click here from n	s any Node that complies with this Rule's i rejects any Node that complies with this Rule's nore information.	¥ 🗃 ▲ System	Object ID Prefix	Range T	уре	H	← 1-1 of 1 → Notes	N 🖻		
Discover Matching Nodes	•	1.3.6.1.4.1	.5528	Include in	n rule					
 Extend Default Beh 	avior (beyond Routers and Switches)									
If enabled, NNMi discover complies with this Rule's c	s any Node that responds to SNMP and riterion. Click here from more information.									
Discover Any SNMP Device										

d. The last rule is for switches and routers. Because NNMi discovers these devices by default, do not specify their object IDs (OID). You only need to specify an IP range.

Auto-Discovery Rule *	
(i) Changes are not committed until the top-level form is saved!	
★ Basics ▲	IP Ranges System Object ID Ranges
Auto-Discovery Rules apply only to Default Tenant.	•
* Name	The System Object ID Range filter is '.1.3.6.1' unless specified otherwise.
* Ordering	
Notes	If you provide one or more System Object ID Ranges and IP Address Ranges, each discovered Node must match one of each type of range. If you provide only one or more System Object ID Ranges (no IP Address Ranges), then 'Discover Included Nodes' must = 'Reject Nodes'.
 Purpose of this Auto-Discovery Rule 	Click here for more information about System Object ID Ranges.
If enabled, NNMi discovers any Node that complies with this Rule's	* ₩ C ¬ * ₩ F ¬
criterion. If disabled, NNMi rejects any Node that complies with this Rule's criterion. Click here from more information.	▲ System Object ID Prefix Range Type Notes
Discover Matching Nodes	
Extend Default Behavior (beyond Routers and Switches) If enabled, NNMi discovers any Node that responds to SNMP and complies with this Rule's criterion. Click here from more information.	
Discover Any SNMP	
If enabled, NNMi discovers any Node that responds to ICMP and complies with this Rule's criterion. Click here from more information.	
Discover Non-SNMP Devices	
Rule Criterion	Total: 0 Selected: 0 Filter: OFF Auto refresh: OFF
Canality the ID Address Deess and/or CNB4D Custom Object IDs that	
	_
 Analysis - Summary - No Objects Selected 	

26.3.4 Adding seeded discovery

Collect from NNM

1. Run the following command to check the accurate list of devices in the NNM database:

ovtopodump > topology.out

Replicate to NNMi

2. From NNM, copy the topology.out (export) file and edit it. Alternatively, re-input the entries in a file for import into NNMi.

For a new file, enter an IP address or host name on each line. Because NNMi automatically determines the subnet mask, you need not specify the subnet mask.

```
10.2.32.201 # comment
10.2.32.202 # comment
1nt03.mycorp.net # comment
1nt02.mycorp.net
10.2.32.205
```

Note

Alternatively, you can also use the NNMi console to add a list of nodes.

3. Run the following command:

^{26.} Upgrading from NNM Version 8 or Earlier

nnmloadseeds.ovpl -f newSeedfile

For details, see the *nnmloadseeds.ovpl Reference Page*.

NNMi immediately begins discovering the devices associated with these seeds, and mounts the existing device profile (as well as node groups such as node groups for status monitoring). NNMi spiral discovery continues. For details about how to check the discovered seed results, see 3.3.3 Checking discovery progress.

^{26.} Upgrading from NNM Version 8 or Earlier

26.4 Phase 3: Migrating status monitoring

In NNM, the netmon process performs status monitoring.

• The netmon process models devices (such as nodes that include interfaces) and applies polling parameters primarily at the node level.

In NNMi, you can apply polling parameters at the node, interface, or address level.

26.4.1 Specifying a polling interval

NNM netmon polling process

Collect from NNM

From the NNM user interface, acquire a polling interval.

NNMi polling process

Replicate to NNMi

The NNMi status monitoring configuration is based on node groups or interface groups (or both).

In the NNMi console, from the **Configuration** workspace, choose **Monitoring** > **Monitoring Configuration**. Click the **Node Settings** tab and set **Fault Polling Interval**.

Monitoring Configuration 30											
C 🖻 関 C											
					_						
 Global Control 	~	Interfac	e Settings	Node Settin	ngs	Default S	ettings				
If disabled, previous device state and status values remain unchanged.											
For more information, click here.		When m	ultiple setting	s are defined	I. NN	Mi applies th	hem accordi	ing to the Or	derina numb	ber (lowest n	umber first).
Fachte Cinte Dolling											
Enable State Polling								ы		1011	
If you do not select Enable State Polling above, NNMI disables monitoring for the following object types and resets the previous states		*			-			м	e 111	J OT 4	M 150
for each.					Ena				Enable	Enable	
					and	Enable	Enable IP	Enable	Node	Physical	Poll
Enable Card Polling		▲ On	Name		Wet	Address	Fault	Fault	Sensor	Sensor	Unconnect
Enable Chassis Polling 🖌					Poll	Polling	Polling	Polling	Fault	Fault	Interfaces
Enable Node Sensor					Nod				roung	roning	
Polling		100	Bouters		~	~		~	~	~	
Enable Physical Sensor		000	1 COULCE &			•	-	•	•		
Poling		200	Networking	nfrastructure	~	*	-	*	*	*	
		300	Microsoft Wi	ndows Syste	~	~	÷	~	-	÷	÷
MMMI monitors each discovered Interface according to the first matching		400	Non-SNMP	Devices	~	*	~	~			
configuration setting (most-specific to least-specific: Interface, Node,											
Default). For more information, click here.											

Node Settings 🗙	C 🔒			
(i) Changes are not o	ommitted until the top-level form is saved!			
a Darias				
For more information, cli	ck here.	^		
Orderlag				
Node Group	Restan			
Easte SNMD and	routers 💌			
Web Polling of Node	 			
 Fault Monitoring 				
For more information, cli	tk here.			
ICMP Fault Monitoring				
Enable Management Address Polling	•			
Enable IP Address Fault Polling				
Fault Monitoring				
Enable Interface Fault Polling	•			
Enable Card Fault Polling	•			
Enable Chassis Fault Polling	•			
Enable Node Sensor Fault Polling	•			
Enable Physical Sensor Fault Polling	✓			
Fault Polling Interval	5.00 Minutes 💌			
For more information, cli	k here.			
Extend the Scope	of Polling Beyond Connected Interfaces			
By default, only connects extend the set of monitor with small node or Interfe	d Interfaces are polled. These settings ed interfaces. It is recommended to use them are Groups.			

26.4.2 Selecting a polling protocol

NNM netmon polling process

Collect from NNM

By default, the netmon process uses ICMP to poll individual addresses (individual addresses are viewed as being the same as interfaces). You can also configure NNM so that the netmon process uses SNMP instead of ICMP depending on the devices (SNMP and ICMP are never used at the same time). To determine whether there is any area in which SNMP is being used, check the following file:

- Windows: %OV_CONF%\netmon.snmpStatus
- Linux: \$OV_CONF/netmon.snmpStatus

NNMi polling process

Replicate to NNMi

In NNMi, sets of node and interfaces are defined as node groups and interface groups, respectively. The polling policy is applied to node groups and interface groups on the **Monitoring Configuration** form.

NNMi polling configuration example

For example, to set polling in a VOIP router set (using SNMP and ping), do the following:

1. Using the **Node Groups** form, create a node group for identifying the VOIP router. Save this form and close the window.

Node Groups 38 Node Group * 20					
6 6 5 8 5					
Basics	Device Filters	Additional Filters	Additional Nodes	Child Node Groups	Status
* Name VOIPRouters					
Calculate Status	When using the li	ke or not like operato	ors, use an * (asterisk)	to match zero or more	characters in a
Status No Status	string and a ? (qu	estion mark) to matc	h exactly one characte	er in a string. Valid exa	mples for
Add to View Filter List	hostname: cisco?	hp.com, cisco*.hp.co	om, ftc??gs??.*.hp.cor	m	
Notes	To create an inclu hostedIPAddress For more information	usive IP address rang between 10.10.1.1 A tion, click here.	ge, use the between op AND 10.10.1.255	perator. Valid example:	
You can filter Node Groups using Device Filters, Additional Filters, Additional Nodes, and Child Node Groups. If you use Device Filters	Filter Editor Attribute	Operator	Value		
and Additional Filters, Nodes must match at least one Device Filter	hostname	-	👻 voip		Append
and the Additional Filters specifications to belong to this Node					Insert
Node Groups always are members of this Node Group. See Help				l	Replace
→ Using the Node Group form.				[Append -
To test your Node Group definition, select File \rightarrow Save, then				[AND
Actions → Node Group Details → Preview Members (Current					OR
Group Only).	hostname = vo	olp		[NOT
NNM iSPI Performance				[EXISTS
Used by NNM iSPI Performance for Metrics and NNM iSPI for				(NOT EXISTS
Traffic.				[Delete
	Filter String hostname = voip				

2. On the Monitoring Configuration form, add new node settings as follows:

Monitoring Configuration 🗙											
6 2 3 5											
Global Control		Interfac	e Settings	ode Settir	ngs	Default S	Settings				
If disabled, previous device state and status values remain unchanged. For more information, click here. Enable State Polling If you do not select Enable State Polling above, NNMi disables monitoring for the following object types and resets the previous states	Î	When m	nultiple settings an	e defined	Ena	Mi applies t	hem accordi	ing to the Ori	dering numb	er (lowest n	umber first).
for each. Enable Card Polling Enable Chassis Polling Enable Node Sensor		▲ Ore	Name		SNN and Wet Poll of Nod	Enable Manageme Address Polling	Enable IP Address Fault Polling	Enable Interface Fault Polling	Enable Node Sensor Fault Polling	Enable Physical Sensor Fault Polling	Poll Unconnect Interfaces
Enable Physical Sensor		100	Routers		~	~	-	~	~	~	-
Polling		200	Networking Infra	structure	*	~	-	~	*	*	-
		300	Microsoft Windo	ws Syste	~	~	-	~	-		
NNMi monitors each discovered Interface according to the first matching configuration setting (most-specific to least-specific: Interface, Node, Default). For more information, click here.		400	Non-SNMP Devi	ices	*	*	*	*		•	

3. After specifying ordering values, choose Quick Find from the Node field.

^{26.} Upgrading from NNM Version 8 or Earlier

Node Settings * 🗶	C 🗎
(i) Changes are not a	committed until the top-level form is saved!
· Basics	
For more information, cl	lick here.
* Ordering	90
 Node Group 	
Enable SNMP and Web Polling of Node	Show Analysis Quick Find
Fault Monitoring	Open * New
For more information, cl	lick here.
ICMP Fault Monitoring	
Enable Management Address Polling	
Enable IP Address Fault Polling	
Fault Monitoring	
Enable Interface Fault Polling	
Enable Card Fault Polling	

4. Select the node group for setting monitoring as follows:

^{26.} Upgrading from NNM Version 8 or Earlier

Quick Find

▲ Name				
Important Nodes				
Microsoft Windows Systems				
Neighbor Connections Filter				
Networking Infrastructure Devices				
Non-SNMP Devices				
Routers				
Switches				
VMware ESX Hosts				
VOIPRouters				
Virtual Machines				
Updated: 8/26/16 10:19:49 AM	Total: 11	Selected: 1	Filter: OFF	Auto refresh: OFF
aparton of the for the form	10101.11			
		Clear	OK	Cancel

5. Select the Enable IP Address Fault Polling check box as shown below. Save the form and close the window.

×

Node Settings * 🛪				
	C 🗎			
(i) Changes are not o	committed until the top-level form	n is saved!		
· Basics				
For more information, cl	ick here.	<u> </u>		
Ordering	90			
Node Group	VOIPRouters	- 🗃 -		
Enable SNMP and Web Polling of Node	•			
+ Fault Monitoring				
For more information, cl	ick here.			
CMP Fault Monitoring				
Enable Management Address Polling	~			
Enable IP Address Fault Polling				
Fault Monitoring				
Enable Interface Fault Polling	✓			
Enable Card Fault Polling				

26.4.3 Setting up important nodes

NNMi provides default node groups for important nodes.

If these important nodes have failed or cannot be reached, NNMi displays SEVERE for the node status and generates a NodeDown incident.

NNM netmon polling process

Collect from NNM

NNM offers no settings for important nodes. You can create new settings for important nodes in NNMi.

NNMi polling process

Replicate to NNMi

- 1. In the NNMi console, from the Configuration workspace, choose **Object Groups** > **Node Groups**.
- 2. Open the Important Nodes node group.
- 3. Add important nodes to the group separately for the host name wildcard, device filter, or certain node as shown below.

a. Add a device filter.

Node Groups 32 Node Group * 32						
♂ ♂ □ □ □ □ □ □						
Basics Name Important Nodes	Device Filters Additional Filters Additional Nodes Child Node Groups Status					
Calculate Status Device Filters enable you to determine Node Group members by Device Category, Vendor, Family, or Device Profile. If you configure multiple Device Filters, Nodes must match at least or of the Device Filters and pass any Additional Filters to belong to this Node Group.						
Notes	*					
Important Nodes such as Edge Routers representing the egress points to local networks from WAN environments, and receive special treatment for root cause analysis. Populate this group as makes sense for your environment.	Device Catego Device Vendor Device Family Device Model Server Hitachi Hitachi HPUX Age					
You can filter Node Groups using Device Filters, Additional Filters, Additional Nodes, and Child Node Groups. If you use Device Filters and Additional Filters, Nodes must match at least one Device Filter and the Additional Filters specifications to belong to this Node Group. Nodes that are specified as Additional Nodes and Child Node Groups <i>always</i> are members of this Node Group. See Help → Using the Node Group form.						
To test your Node Group definition, select File \rightarrow Save, then Actions \rightarrow Node Group Details \rightarrow Preview Members (Current Group Only).						

b. Add certain nodes. Save the form and close the window.

Node Groups 30 Node Group * 20					
6 6 5 8 5					
Basics Name Important Nodes Calculate Status No Status Add to View Filter List Notes Important Nodes such as Edge Routers representing the egress points to local networks from WAN environments, and receive special treatment for root cause analysis. Populate this group as makes sense for your environment.	^	Device Filters	Additional Filters	Additional Nodes	Child Node Groups Status egardless of any filters. ◆ 1-2 of 2 → N □□
You can filter Node Groups using Device Filters, Additional Filters, Additional Nodes, and Child Node Groups. If you use Device Filters and Additional Filters, Nodes must match at least one Device Filter and the Additional Filters specifications to belong to this Node Group. Nodes that are specified as Additional Nodes and Child Node Groups always are members of this Node Group. See Help → Using the Node Group form. To test your Node Group definition, select File → Save, then Actions → Node Group Details → Preview Members (Current Group Only).					

26.4.4 Excluding objects from state polling

In NNM, nearly all activities that stop monitoring of nodes or interfaces (putting them in the **Not Managed** state) are set manually through the NNM user interface.

NNMi simplifies the process for excluding objects from management. You can match the default in the new operation to the operation that used to be performed manually (for example, polling only uplinks). However, using node groups and interface groups to manage configurations simplifies automatic configuration updating.

Sometimes it is necessary to mark nodes or interface as **Not Managed**. You can set the management mode for individual nodes on the **Node** form as follows:



You can set the management mode for individual interfaces on the Interface form as follows:

Interfaces (All Attributes	i) 30 Interface x									
0 0 D 13	0									
· Basics		~	< General	IP Addresses	Ports	VLAN Ports	WAN Connections	Capabilities	C >	~
Name	Fa0/12		 System Pl 	roperties						
Status	Normal		ifName	Fa0/12						
Management Mode	Managed		ifAlias							
Direct Management	Inherited 🕶		ifDescr FastEthernet0/12							
	Choose One		ifIndex	12						
Hosted On Node	Inherited	11	ifSpeed	100 Mbps						
Physical Address	Not Managed		ifType	ethemetC	smacd				- 👘 ·	-
Laver 2 Connection	Out of Service	1e103003/Ea0/1								-
cayer 2 connection	arigio		 Input/Outp 	put Speed						
 Interface State 			Input Speed	10000000	00					
Administrative State	Up		Output Speed	10000000	00					
Operational State	Up									
State Last Modified	August 31, 2016 1:55:39 PM JST									

26.5 Phase 4: Migrating event configuration and event reduction

NNM uses the expanded SNMPv2 format to analyze all sources of received events (traps from managed devices, internal process communication, and transferred events). Each event has an event object identifier, a name, and configuration parameters.

NNMi processes the sources of various events differently. The traps from devices are in the SNMPv2 format. Furthermore, NNMi's internal process communication uses a new (non-trap) mechanism to substantially improve overall performance. NNMi does not generate the no format in trapd.conf message in response to unrecognized events. Unrecognized messages are now discarded as the default.

Because the following event configuration areas are not used in NNMi, they cannot be migrated:

• Several types of configuration element correlation process: suppress, enhance, transient, and multisource

26.5.1 Displaying traps from devices

You can configure NNMi to display traps from devices in a way similar to that used in the NNM environment.

NNMi includes many default settings for the general SNMP traps and vendor traps packaged in NNM. You can update NNMi by updating these traps.

For the list of variables that can be used in messages and automatic actions, see Valid Parameters for Configuring Incident Messages (SNMP Trap Incident) and Valid Parameters for Configuring Incident Actions (SNMP Trap Incident) in NNMi Help.

Collect from NNM

1. Check whether the NNM configuration contains customized traps.

Pay attention to the customizing that has been done in relation to categories, importance, displayed messages, and automatic actions.

Replicate to NNMi

- 2. Download the vendor MIB file to the NNMi management server.
- 3. For each MIB, run the following command:

```
nnmloadmib.ovpl -load mibFile
nnmincidentcfg.ovpl -loadTraps mibModule
```

• To determine which MIBs have already been loaded, run the following command:

nnmloadmib.ovpl -list

For details, see the nnmincidentcfg.ovpl Reference Page and the nnmloadmib.ovpl Reference Page.

Note

These steps only load TRAP-TYPE and NOTIFICATION-TYPE MIB entries, and NNMi does not use other MIB variables.

4. In the NNMi console, from the **Configuration** workspace, choose **Incidents** > **SNMP Trap Configuration**.

Network Node Manager i	Elle View Icols	Agtions Help				User N	iame: administrator NNMi	Role: Administ	trator	Sign Out
M Dashboards	SNMP Trap Configurations	-								
e [£] Incident Management								1 10 -172	N N	
📥 Topology Maps	G * 3 •	• 1 5 ■					N 4	1-190172	FI	
C Monitoring	▲ Name	SNMP Object ID	Enabled	Root Cause	Deduplicat Enabled	Rate Enabled	Set Cal Far Author	Format		
A Troubleshooting	ArcSightEvent	.1.3.6.1.4.1.11937.0.1				-	O 🏨 🗞 HP Art Sight	\$1.3.6.1.4.1		
E Inventory	BGPBackwardTransition	.1.3.6.1.2.1.15.0.2				-	A 🔹 📇 HP Network N	BGP Backwa		^
A Management Mode	BGPEstablished	1.3.6.1.2.1.15.0.1					🔿 🏨 🗮 HP Network N	BGP Estable		
A Incident Browsing	CempMemBufferNotify	.1.3.6.1.4.1.9.9.221.0.1	~				🛇 🌟 🖠 HP Network N	Memory buff		
% Integration Module Configuration	CiscoChassisAlarmOff	.1.3.6.1.4.1.9.5.0.6	-		~		📀 🏨 🚟 HP Network N	Cisco Chass		
F Configuration	CiscoChassisAlarmOn	.1.3.6.1.4.1.9.5.0.5	1.0		~	-	A 🕸 📇 HP Network N	Cisco Chass		
Communication Configuration	CiscoChassisChangeNotificat	.1.3.6.1.4.1.9.5.11.2.0.2	-	-	~	-	A 🔹 👼 HP Network N	Cisco Chass		
Discovery	CiscoColdStart	1361631151136141	~			~	🔿 🏨 🌆 HP Network N	Cisco Agent		
Monitoring	CiscoDemandNeighborLayer2	1.3.6.1.4.1.9.9.26.2.0.3			~	-	A 🔹 🚟 HP Network N	Demand Nei		
🕆 🗁 Incidents	CiscoEnvMonFanNotification	1361419913304	~			-	😋 🏨 🔰 HP Network N	Fan state chi		
Incident Configuration	CiscoEnvMonFanStatusChan	1361419913308	~				🔕 🎄 🛊 HP Network N	Fan state ch		
SNMP Trap Configurations	CiscoEnvMonRedundantSupp	1361419913305	~			-	A HP Network N	Power supply		
Syslog Message Configurations	CiscoEnvMonSuppStatusCha	1361419913309	~				O 👾 🔒 HP Network N	Power supply		
IIII Management Event Configurate	CiscoEnvMonTemoStatusCha	1361419913307					G - H I HD Network N	Temperature		
Custom Correlation Configuration	CiscoEnvMonTemperatureNo	1361419913303	-				O de di HD Network N	Temperature		
Trap Server	CiscoEm Alon) (ol Status Chan	1361410013306					C -th d HD Nature N	Voltage state		
Status Configuration	CiscoEn Alora Joltana Mattina d	1361410013303	*				C die d HD Natural N	Voltage state		
Global Network Management	CiscoEnviron/oitagervotricat	1.1.3.6.1.4.1.9.9.13.3.0.2				-	C th We have the	voltage state		
User Interface	CiscoPRUInserted	.1.3.6.1.4.1.9.9.117.2.0.3	*			-	C A BURNEWOK N	PRU with hai		
Security	CiscoPRURemoved	.1.3.6.1.4.1.9.9.117.2.0.4	~	•		•	Se alle HP Network N	PRU with nai		
MIBs	Updated: 8/26/16 10:45:31 AM		Tol	al: 72	Selected	t: 0	Filter: OFF		Auto refres	h: OFF
Device Profiles					-					
S = 10	 Analysis - Summary - No 	Objects Selected								

5. Customize the display so that the trap display matches the display in NNM.

You can create categories as needed on the SNMP Trap Configuration form.

^{26.} Upgrading from NNM Version 8 or Earlier
SNMP Trap Configura	tions sc	SNMP Trap	Configuration	•								
0 0 0	9	C 🗎										
 Basics 					<	Interface	Settings	Node Settings	Suppression	Enrichment	Damp >	~
For information about	troublesh	ooting Incidents	s, click here.		•							
Name The SNMP Object ID character (*) that mus NNMi permits wildcarr (private MIBs). Click f	SNM (OID) attri t appear a ts only in ere for mo	PLinkDown ibute accepts o at the end of the OIDs beginning ore information.	ne wildcard e OID specified. g with .1.3.6.1.4		NNM Obje oven inclu	Ai enables y oct based or ride any oth iding those	ou to apply the Source er Suppres configured	v a Suppress, Enric e Object's participa ss, Enrich, Damper on the Node Settin	h, Dampen, or Ac tion in an Interfac n, or Action config gs tab.	tion configuratio e Group. Interfa uration settings t	n to a Source ce Settings for this Incide	nt,
-						Interface C	Group Or	dering Enabled				_
SNMP Object ID	.1.3.6	.1.6.3.1.1.5.3										
Enabled	 Image: A start of the start of											
Root Cause												
 Category 	Fault		- 🖷 🗸									
* Family	Interf	ace	- 🐨 🕶									
 Severity 	Critic	al 💌										
Specify how the Incide To include Incident int \$(variable_name). Se parameters or Custon click here.	ent messa formation i lect these a Incident	ge appears in t in the message variables from attributes. For r	the Incident view. Luse a set of valid more information,									
 Message Format 												
Agent Interface Dow	n (linkDov	wn Trap) on inte	erface \$1	J				Total	0 Selected: 0 I	Filter OFF Auto	o refresh: OF	F
Description				~				, otal.		AUG. OFF POR	o tonoon. Of	
12550000						_						
Analysis - SNMP	rap Con	figuration Sum	mary : SNMPLin	kDowr	n - SN	IMP Object	ID: .1.3.6.1	.6.3.1.1.5.3 Mess	age Format: Ager	nt Interface Dov	vn (linkDowr	Trap)

NNMi enhancement

on interface \$1

- 6. (Optional) Set up Family as the default in addition to the default settings of Severity, Category, and Message Format.
- 7. (Optional) Classify traps as root causes so that traps are displayed in the Open Root Cause Incidents view.

26.5.2 Customizing the display of management events generated by NNMi

In NNMi, event configuration is simple. The NNMi Causal Engine generates root causes that are more concise than in NNM.

Change the incidents generated by NNMi to make them appear similar to the NNM alarms. For example, you can customize the NNMi NodeDown incident message to appear similar to the NNM NodeDown alarm message:

Collect from NNM

1. In NNM, determine whether event configuration has been customized.

Replicate to NNMi

2. In the NNMi console, from the **Configuration** workspace, choose **Incidents** > **Management Event Configurations**.

^{26.} Upgrading from NNM Version 8 or Earlier

- 3. Look for a new incident configuration using a name instead of an event number.
- 4. (Optional) To customize the event display so that it matches the event display in NNM, create a category on the management event configuration form.
- 5. You can set up Family as the default in addition to the default settings of Severity, Category, and Message Format.

26.5.3 Setting up blocking, ignoring, or disabling of traps

NNM is equipped to perform various levels of event processing, such as the following:

- Blocks traps when they come into ovtrapd.
- Processes traps or events labeled IGNORE, but does not save or display them.
- Saves and processes (correlates) events labeled LOGONLY, but does not display them.
- Saves events in categories, processes and displays them.
- Displays traps arriving without settings as No format in trapd.conf for . . . in the alarm browser, and saves them in the database.

NNMi has a much simpler method. It does not save, process, or display *disabled* events or traps. NNMi fully saves, processes, and displays *enabled* events or traps. Events that are not set in NNMi are blocked.

Collect from NNM

- 1. Determine whether customization that ignores traps or customization that sets traps in LOGONLY is used.
- 2. Check whether NNM uses a trap-filtering mechanism (ovtrapd.conf, which was new in NNM 08-00).

 Replicate to NNMi

 Replicate to NNMi
- 3. In the NNMi console, from the **Configuration** workspace, choose **Incident Configuration**. Find events you do not want to receive or display, and clear the **Enabled** check box for these events.
- 4. To block traps from certain IP addresses, edit the following file and update NNMi by using the trap-filtering information from NNM:
 - Windows: %NnmDataDir%shared\nnm\conf\nnmtrapd.conf
 - Linux: \$NnmDataDir/shared/nnm/conf/nnmtrapd.conf
- 5. Enable trap blocking by running the nnmtrapconfig.ovpl command with a trap-blocking rate and a threshold value specified.

For details about how to use this command, see the nnmtrapconfig.ovpl Reference Page.

26.5.4 Setting up automatic actions

Collect from NNM

1. Determine the automatic actions that are set for NNM.

Replicate to NNMi

2. Copy the action scripts of the NNM management station to the NNMi management server. In this case, the file location is not important.

^{26.} Upgrading from NNM Version 8 or Earlier

- 3. In the NNMi console, from the **Configuration** workspace, choose **Incident Configuration**.
- 4. For each NNM event for which an automatic action is set, configure the corresponding NNMi incident with that action (Action Configuration tab).

To enable an action, you must select the **Enabled** check box.

5. To match the NNMi actions with NNM actions, set Lifecycle State to Registered.

NNMi enhancement

- 6. Pay attention to the following NNMi configurations:
 - You can configure multiple automatic processes that are generated when events arrive.
 - You can set up one or more additional processes for each lifecycle state (which can be In Progress, Completed, or Closed).
 - You can pass on more incident attributes to commands than was possible with NNM.
 - Because there is no need to register a command in another configuration file before NNMi executes the command, the procedure has become simpler.

26.5.5 Setting up additional (manual) actions

NNM provides operator actions or additional actions that can be used from the alarm browser menu. You can also simulate NNM actions by using the URL actions that can be utilized from the NNMi console.

Collect from NNM

1. Determine the custom operator actions being used in NNM.

Replicate to NNMi

- 2. Determine the method of migrating these custom actions so that they can be utilized as URL actions.
- 3. In the NNMi console, from the **Configuration** workspace, choose **User Interface** > **Menu Items**.
- 4. Click New.
- 5. Prepare Menu Item Label, Unique Key, Order, Selection Type, as well as Menu Item Contexts for actions.

26.5.6 Event correlation: Event repetition

When duplicating events, NNM uses either the first or the last event as the parent.

In NNMi, a new parent is created, which is displayed when you choose **All Incidents** from the **Incident Browsing** workspace. The original event is also displayed in the view that has been set up.

Collect from NNM

- 1. Check whether RepeatedEvents correlation is used in NNM.
- 2. Check whether Repeated correlation is used in NNM.
- 3. Check whether duplication is used (dedup.conf file).

Replicate to NNMi

4. In the NNMi console, from the Configuration workspace, choose Incident Configuration.

^{26.} Upgrading from NNM Version 8 or Earlier

- 5. Open the event that is to be duplicated.
- 6. Click the **Deduplication** tab, enable deduplication, select a new parent event, and then define criteria for matching.

Duplication	in NNMi has no time	limit.							
SNMP Trap Configura	ations 🗶 SNMP Trap Configuration 🗶								
	9 🛛 🤇 📋								
 Basics 		< ment	Dampening	Deduplication	Rate	Actions	Forward to Glo	ba 🔪 🗸	
For information about	t troubleshooting Incidents, click here.								
Name The SNMP Object ID character (*) that mus NNMi permits wildcar	SNMPColdStart (OID) attribute accepts one wildcard st appear at the end of the OID specified. rts only in OIDs beginning with 1 3 6 1 4	Deduplication duplicate. Du number of du incremented	n determines w plicate Inciden plicates gener on the Duplica	hat values NNMi s ts are listed under ated. This value is te Correlation Incid	hould mai a Duplica captured lent.	ich to detect te Correlatio as the Dupli	when an Incident n Incident. NNMi cate Count attribu	t is a tracks the ite. It is	
(private MIBs). Click	here for more information.	Enabled	•	•					_
		Count	1	0					-
SNMP Object ID	.1.3.6.1.6.3.1.1.5.1	Hours	4						-
Enabled		Minutes	0						1
Root Cause		Seconds	0						1
* Category	Fault 💌 🖼 🗸	Parent Incid	lent D	uplicateCorrelation	1			- 🖼 -	1
* Family	Node 💌 🖉 🗸	* Comparisor	Criteria	lame CIA 👻					
* Severity	Normal 👻	- Dedun	lication Comp	arison Parameters					
Specify how the Incid To include Incident in \$(variable_name). Se parameters or Custor click here.	ent message appears in the Incident view. formation in the message use elect these variables from a set of valid m Incident attributes. For more information,	★ 🗃 ▲Parame cia.addres	C S ter Value s	1		₩ ←	<u>1-1 of 1</u> →	H 🚍	0
* Message Format									
Agent Up with Poss	ible Changes (coldStart Trap)								_
Description	`	·		Total	1 Sele	ted: 0 Filt	er: OFF Auto re	efresh: OFF	
			_						

26.5.7 Event correlation: Rate calculation

When duplicating events, NNM uses either the first or the last event as the parent.

In NNMi, a new parent is created, which is displayed when you choose **All Incidents** from the **Incident Browsing** workspace. The original event is also displayed in the view that has been set up. In NNMi, the rate actions have been configured to match those in the Regular Time window in NNM.

Collect from NNM

1. Check whether rate correlation is used in NNM.

Replicate to NNMi

- 2. In the NNMi console, from the **Configuration** workspace, choose **Incidents** > **Management Event Configurations**.
- 3. Open the event identifier to be counted.
- 4. Click the **Rate** tab and perform the following:
 - a. Choose Enabled to enable monitoring.
 - **b.** Specify a counting range.
 - c. Specify a time range (using the Hours, Minutes, and Seconds fields).

^{26.} Upgrading from NNM Version 8 or Earlier

d. Select a new parent event (Correlation Incident Configuration).

e. Define Comparison Conditions.

For details, see Management Event Form in NNMi Help.

26.5.8 Event correlation: Pairwise cancellation

In NNMi, there is no specific time limit for cancellation.

Collect from NNM

- 1. Check whether PairWise (pair event) correlation is performed in NNM.
- 2. Check whether transition state correlator is used in NNM.

Replicate to NNMi

- 3. In the NNMi console, from the **Configuration** workspace, choose **Incidents** > **Pairwise Configurations**.
- 4. Either select an existing pair or click New.
- 5. Set up the event identifiers to be paired and the criteria for matching. For details, see *Configuring Incidents* in NNMi Help.

26.5.9 Event correlation: ScheduledMaintenance

With NNMi, you can suppress the monitoring of unusable nodes. To do this, use the **Out of Service** mode. Refer to 21.26 Scheduling outages about how to configure **Out of Service** maintenance in advance.

Note

NNMi suppresses SNMP traps sent from devices that are in the Out of Service mode.

If your organization is using ScheduledMaintenance correlation, you can use a list of systems that will go offline together.

Collect from NNM

1. Check whether ScheduledMaintenance correlation is used in NNM.

Replicate to NNMi

- 2. In the NNMi console, from the Configuration workspace, choose Object Groups > Node Groups.
- 3. Create a node group for each node set in the NNM maintenance list. Configure the node groups so that they can be used as view filters.
- 4. For maintenance, from the Inventory workspace in the NNMi console, choose Nodes.
- 5. To filter the view for a given node group, use the **Node Group Filter Configuration** selector at the top of the window.
- 6. Select all nodes, and then choose Actions, Management Mode, and Out of Service.
- 7. After the maintenance has been completed, select a node, and then choose **Actions**, **Management Mode**, and **Manage**.

^{26.} Upgrading from NNM Version 8 or Earlier

27

Migrating NNMi from an HP-UX or Solaris Operating System

If you are running NNMi 10-50-02 or later in an HP-UX or Solaris operating system, follow the procedure described in this chapter to migrate NNMi. NNMi 11-00 does not support HP-UX or Solaris operating systems. You must change the operating system to a supported Linux operating system before migrating to NNMi 11-00. For details about the supported operating systems, see Preface.

27.1 Changing NNMi from HP-UX or Solaris to Linux

To complete the following steps, you must have NNMi latest patch of 10-50 running on an HP-UX or Solaris server.

Verify in the window at **Help** > **About Network Node Manager i** that the NNMi version is latest patch of 10-50. Do not proceed if the version is not latest patch of 10-50; in such a case, you must first upgrade to NNMi latest patch of 10-50.

The procedure below explains how to change an NNMi management server running NNMi latest patch of 10-50 from an HP-UX or Solaris operating system.

If you are running NNMi on Application Failover environment, see 27.2 Changing NNMi from HP-UX or Solaris to Linux on Application Failover environment.

If you are running NNMi on Global Network Management environment, see 27.3 Changing NNMi from HP-UX or Solaris to Linux on Application Failover environment on the global manager and regional managers.

If you are running NNMi on High-Availability environment, see 27.4 Changing NNMi from HP-UX or Solaris to Linux in a High Availability Cluster.

This procedure uses the following two servers:

- Server A is the current NNMi management server running HP-UX or Solaris.
- Server B is a new NNMi management server that will be running RHEL 6.

Server B cannot employ the same hardware configuration as the current Server A.

To change the NNMi management server:

1. On Server A, run backup processing.

Make a full backup of NNMi using the following command:

nnmbackup.ovpl -type online -scope all -target temporary_location

You must use the online option. For details, see the nnmbackup.ovpl Reference Page.

- 2. On Server B, install RHEL 6.
- 3. On Server B, install NNMi.

Install NNMi on Server B.

You must install at the same patch level that was in effect on NNMi Server A at the time the backup was made in step 1.

Important

You cannot back up with one version of NNMi and restore to a different version of NNMi. Therefore, Server B must be the identical NNMi version, including patch level, as Server A.

Note

If you are using an Oracle database, select **Secondary Server Installation** during the installation process.

4. Save the port settings.

^{27.} Migrating NNMi from an HP-UX or Solaris Operating System

During the NNMi installation on Server B, the installation script might assign ports that differ from the port settings in the Server A configuration. This might create port conflicts on Server B when the configuration is restored. To remedy this, do the following:

a. On Server B, navigate to the \$NNM_CONF/nnm/props/ directory.

b. On Server B, copy the nms-local.properties file to nms-local.properties.save in a temporary location.

5. On Server B, restore the backup.

Use the following procedure to restore to Server B the backup made in step 1:

a. On Server A, copy the backup made in step 1 to a temporary location on Server B.

b. On Server B, complete a full NNMi restore by running the following command:

nnmrestore.ovpl -force -source temporary location

For details about the command options to be used, see 20. NNMi Backup and Restore Tools and the *nnmrestore.ovpl Reference Page*.

Important

Use the command options that match the backup you completed in step 1.

6. Resolve any port conflicts.

On Server B, compare the nms-local.properties.save file from the temporary location to the nmslocal.properties file located in the \$NNM_CONF/nnm/props/ directory.

Resolve any port conflicts, making changes to the nms-local.properties file located in the directory noted above. Make sure to keep the values for jboss.http.port (NNMi Web server port) and jboss.https.port (NNMi HTTPS Web server port) that were set during NNMi installation on Server B.

7. Restart the NNMi management server.

Restart the NNMi management server by running the following commands on Server B:

ovstop ovstart

8. Apply license keys on Server B.

If the IP address of Server B is different from the IP address of Server A, obtain and install new NNMi license keys. See 22.4 Changing the IP address of a stand-alone NNMi management server.

If the IP address of Server B is the same as the IP address of Server A, apply to Server B the license keys that apply to Server A.

9. On Server B, perform overwrite installation of NNMi 11-00.

For details about how to perform overwrite installation of NNMi 11-00, see the *Release Notes*. Before you install NNMi 11-00, you must obtain 11-00 license keys. For details, see 2.3 Licensing NNMi.

27. Migrating NNMi from an HP-UX or Solaris Operating System

27.2 Changing NNMi from HP-UX or Solaris to Linux on Application Failover environment

This procedure uses the following four servers:

- Server A is the current active NNMi management server running on HP-UX or Solaris.
- Server a is the current standby NNMi management server running on HP-UX or Solaris.
- Server B is a new active NNMi management server that will be running on RHEL 6.
- Server b is a new standby NNMi management server that will be running on RHEL 6.

Both Server B and b cannot emply the same hardware configuration as the current Server A, a.

1. Disable application failover on Server A, a.

Refer to 16.6 Disabling application failover described in NNMi 10-50 Setup Guide.#

#: Setup Guide refers to one of the following manuals.

- JP1/Cm2/Network Node Manager i Setup Guide (3021-3-242-20)

- Job Management Partner 1/Consolidated Management 2/Network Node Manager i Setup Guide (3021-3-343-20)

2. On Server A, run backup processing.

Make a full backup of NNMi using the following command:

nnmbackup.ovpl -type online -scope all -target temporary_location

You must use the online option. For details, see the nnmbackup.ovpl Reference Page.

- 3. On Server B and b, run step 2 and 3 described in 27.1 Changing NNMi from HP-UX or Solaris to Linux in NNMi 11-00 Setup Guide.
- 4. On server B, run step 4 through step 7 described in 27.1 Changing NNMi from HP-UX or Solaris to Linux in NNMi 11-00 Setup Guide.
- 5. On Server B and b, run step 8 and 9 described in 27.1 Changing NNMi from HP-UX or Solaris to Linux in NNMi 11-00 Setup Guide.
- Configure application failover between Server B and b. Refer to 18.3 Configuring NNMi for application failover.

27. Migrating NNMi from an HP-UX or Solaris Operating System

27.3 Changing NNMi from HP-UX or Solaris to Linux on Application Failover environment on the global manager and regional managers

This procedure uses the following four servers:

- Server A is the current global manager running on HP-UX or Solaris.
- Server a is the current regional manager running on HP-UX or Solaris.
- Server B is a new active NNMi management server that will be running on RHEL 6.
- Server b is a new standby NNMi management server that will be running on RHEL 6.

Both Server B and b cannot emply the same hardware configuration as the current Server A, a.

Important

Run all of the following steps on all reginal managers if there is more than one regional managers.

- On Server A, disconnect communication between the global manager and regional managers. Refer to 13.9 Disconnecting communication between global1 and regional1 in NNMi 10-50 Setup Guide[#].
 - #: Setup Guide refers to one of the following manuals.
 - JP1/Cm2/Network Node Manager i Setup Guide (3021-3-242-20)

- Job Management Partner 1/Consolidated Management 2/Network Node Manager i Setup Guide (3021-3-343-20)

- 2. On Server A and a, run step 1 described in 27.1 Changing NNMi from HP-UX or Solaris to Linux in NNMi 11-00 Setup Guide.
- 3. On Server B and b, run step 2 through 9 described in 27.1 Changing NNMi from HP-UX or Solaris to Linux in NNMi 11-00 Setup Guide.
- 4. Connect communication between Server B and b.

Refer to 15.6 Connecting a global manager with a regional manager described in NNMi 11-00 Setup Guide.

^{27.} Migrating NNMi from an HP-UX or Solaris Operating System

27.4 Changing NNMi from HP-UX or Solaris to Linux in a High Availability Cluster

You can change NNMi from HP-UX or Solaris in a High Availability Cluster to Linux in a High Availability Cluster. On Linux, you can use either Veritas Cluster Server or HA monitor.

This procedure uses the following four servers:

- Server A is the current active cluster node running on HP-UX or Solaris.
- Server a is the current passive cluster node running on HP-UX or Solaris.
- Server B is a new primary cluster node that will be running on RHEL 6.
- Server b is a new secondary cluster node that will be running on RHEL 6.

Both Server B and b cannot emply the same hardware configuration as the current Server A, a.

1. On Server A and a, unconfigure NNMi from an HA cluster.

Refer to 17.7 Unconfiguring NNMi from an HA cluster described in NNMi 10-50 Setup Guide.

Make sure to run the following steps described in Setup Guide.[#] Otherwise DB is NOT transferred.

#: Setup Guide refers to one of the following manuals.

- JP1/Cm2/Network Node Manager i Setup Guide (3021-3-242-20)

- Job Management Partner 1/Consolidated Management 2/Network Node Manager i Setup Guide (3021-3-343-20)

2. On Server A, run backup processing.

Make a full backup of NNMi using the following command on current active cluster node:

nnmbackup.ovpl -type online -scope all -target temporary_location

You must use the online option. For details, see the nnmbackup.ovpl Reference Page.

- 3. On Server B and b, run step 2 and step 3 described in 27.1 Changing NNMi from HP-UX or Solaris to Linux in NNMi 11-00 Setup Guide.
- 4. On Server B, run step 4 through step 7 described in 27.1 Changing NNMi from HP-UX or Solaris to Linux in NNMi 11-00 Setup Guide.
- 5. On Server B and b, run step 8 and step 9 described in 27.1 Changing NNMi from HP-UX or Solaris to Linux in NNMi 11-00 Setup Guide.
- 6. On Server B, delete the following line in /var/opt/OV/shared/nnm/conf/ov.conf

```
# HA_EVENTDB_DIR=<mount point of shared disk>/NNM/dataDir/shared/nnm/databases/eve
ntdb
```

7. Configuring HA between B and b.

Refer to 19.4 Configuring HA described in NNMi 11-00 Setup Guide.

Important

For HA Monitor, you perform the configuration procedure without using nnmhaconfigure.ovpl. For details about the configuration procedure, see the Release Notes.

^{27.} Migrating NNMi from an HP-UX or Solaris Operating System



NNMi Northbound Interface

Job Management Partner 1/Consolidated Management 2/Network Node Manager i (NNMi) provides the NNMi Northbound interface for forwarding NNMi incidents to any application that can receive SNMPv2c traps. For each NNMi management server, you can implement the NNMi Northbound interface to multiple Northbound applications, each configured separately. This chapter explains how to configure NNMi to forward NNMi incidents to desired Northbound applications. For details about a specific Northbound application, see the application documentation. The chapter also discusses integration with different Northbound applications.

28.1 Overview of the NNMi Northbound interface

The following provides an overview of the NNMi Northbound interface:

• The NNMi Northbound interface forwards NNMi management events as SNMPv2c traps to a Northbound application.

The Northbound application might filter, act on, and show the NNMi traps. The Northbound application might also provide tools for accessing the NNMi console in the context of an NNMi trap.

- The NNMi Northbound interface can send incident lifecycle state change notifications, incident correlation notifications, and incident deletion notifications to the Northbound application. In this way, the Northbound application can replicate the results of NNMi causal analysis.
- The NNMi Northbound interface can also forward to the Northbound application the SNMP traps that NNMi receives.
- The NNMi Northbound interface enables event consolidation in a third-party or custom event consolidator.
- The NNMi Northbound interface enriches events with information that can be used to integrate other applications with NNMi.

This chapter uses the following terms:

- Northbound application: Any application that can receive and process SNMPv2c traps.
- Trap-receiving component: The portion of a Northbound application that receives SNMP traps. Some applications include a separately installable component that receives SNMP traps and forwards them to another component for processing.

For any Northbound application that does not include such a component, *trap-receiving component* is synonymous with *Northbound application*.

- NNMi Northbound interface: The NNMi functionality that forwards NNMi incidents as SNMPv2c traps to a Northbound application.
- Northbound destination: A configuration of the NNMi Northbound interface that defines the connection to the trapreceiving component of a Northbound application and specifies the types of traps that NNMi will send to that Northbound application.

28.2 Enabling the NNMi Northbound interface

NNMi does not limit the amount of information sent in an SNMP trap using UDP. If any network hardware in the transmission path cannot handle the size of the trap data, or if network traffic is heavy, the trap might be lost. For this reason, we recommend that you install the trap-receiving component of the Northbound application on the NNMi management server. The Northbound application is responsible for ensuring reliable information transfer.

To enable the NNMi Northbound interface:

- 1. If necessary, configure the Northbound application to understand the NNMi trap definitions.
- 2. On the NNMi management server, configure NNMi incident forwarding:

a. In the NNMi console, open the NNMi Northbound Interface Destinations form (Integration Module Configuration > Northbound Interface), and then click New.

If you have selected an available destination, click Reset to make the New button available.

- **b.** Select the **Enabled** check box to make the remaining fields on the form available.
- c. Enter the information for connecting to the Northbound application.

For details about these fields, see 28.8.1 NNMi Northbound application connection parameters.

d. Specify the sending options and incident filter for the content to send to the Northbound application.

For details about these fields, see 28.8.2 NNMi Northbound interface integration content.

e. Click Submit at the bottom of the form.

A new window opens that displays a status message. If the message indicates a problem with the settings, click **Return** and adjust the values as suggested in the error message text.

3. (Optional) Create contextual interaction with NNMi by creating URLs that provide access to NNMi views from the Northbound application.

NNMi does not limit the amount of information sent in an SNMP trap using UDP. If any network hardware in the transmission path cannot handle the size of the trap data, or if network traffic is heavy, the trap might be lost. For this reason, we recommend that you install the trap-receiving component of the Northbound application on the NNMi management server. The Northbound application is responsible for ensuring reliable information transfer.

For details, in the NNMi console, click Help > NNMi Documentation Library > Integrate NNMi Elsewhere with URLs.

28.3 Using the NNMi Northbound interface

When the NNMi Northbound interface is enabled, the Northbound destination determines the information that NNMi sends to a Northbound application. Configure the Northbound application to show and interpret the forwarded traps, as appropriate in your network environment. For details about the contents and format of the traps that NNMi sends to a Northbound application, see the hp-nnmi-nbi.mib and hp-nnmi-registrations.mib files.

NNMi sends only one copy of each management event, SNMP trap, or notification trap to a Northbound destination. NNMi does not queue traps. If the trap-receiving component of a Northbound application is unavailable when NNMi forwards a trap, the trap is lost.

This section describes the types of traps the integration can send. For details about setting the content configuration, see 28.8.2 NNMi Northbound interface integration content.

28.3.1 Incident forwarding

(1) Management events

When the Northbound destination includes management events, NNMi forwards each management event incident to the Northbound application when that incident changes to the **Registered** lifecycle state.

The OID of the forwarded management event is the SNMP object ID on the **Management Event Configuration** form in the NNMi console. NNMi forwards all custom management events with the OID 1.3.6.1.4.1.11.2.17.19.2.0.9999.

(2) Third-party SNMP traps

When the Northbound destination includes third-party SNMP traps, NNMi forwards each incoming SNMPv1, v2c, or v3 format trap to the Northbound application when the associated incident changes to the **Registered** lifecycle state. NNMi preserves the original trap varbinds in order (as defined in the MIB) and appends the NNMi-specific varbinds to the message payload. If the original trap does not contain all the defined varbinds, NNMi pads NULL values for the missing varbinds. If the MIB is not loaded in NNMi, only the NNMi specific varbinds are appended to the trap, which is then forwarded.

Note the following about third-party SNMP traps:

- Because NNMi reconstructs a trap from its SNMP trap incident, the forwarded trap is in SNMPv2c format regardless of the format of the original trap when NNMi received it.
- The forwarded SNMP trap shows the NNMi management server as the source object. To determine the original source object, examine the values of the (n + 21)th varbind, nnmilncidentSourceNodeHostname

(1.3.6.1.4.1.11.2.17.19.2.2.21), and the (n + 24)th varbind, nnmiIncidentSourceNodeMgmtAddr (1.3.6.1.4.1.11.2.17.19.2.2.24), where *n* is the number of varbinds defined for the trap in the MIB.

If any of the devices that NNMi manages also send traps to the Northbound application, the Northbound application must manage the duplicate device traps.

For a comparison of trap forwarding mechanisms, see 8.1.2 Trap and incident forwarding.

^{28.} NNMi Northbound Interface

JP1/Network Node Manager i Setup Guide

28.3.2 Incident lifecycle state change notifications

The information in this section varies with the selections made to the **Sending Options** on the **NNMi Northbound Interface Destination** page.

(1) Enhanced closed traps

When the Northbound destination includes enhanced closed notifications, NNMi sends an nnmiEvClosed (1.3.6.1.4.1.11.2.17.19.2.0.1000) trap to the Northbound application when the lifecycle state of an incident changes to CLOSED in NNMi. The nnmiEvClosed trap includes much of the data from the original incident. The previous lifecycle state value is not included.

The nnmiEvClosed trap identifies the original incident in the sixth varbind, nnmiIncidentUuid (1.3.6.1.4.1.11.2.17.19.2.2.6).

(2) State change traps

When the Northbound destination includes lifecycle state changed notifications, NNMi sends a nnmiEvLifecycleStateChanged (1.3.6.1.4.1.11.2.17.19.2.0.1001) trap to the Northbound application when the lifecycle state of an incident changes to the IN PROGESS, COMPLETED, or CLOSED lifecycle state in NNMi. The Northbound application can associate the nnmiEvLifecycleStateChanged with the original incident.

The nnmiEvLifecycleStateChanged trap identifies the original incident and the lifecycle state change in the following varbinds:

- nnmiIncidentUuid, the sixth varbind (1.3.6.1.4.1.11.2.17.19.2.2.6)
 This value matches the value of the sixth varbind in a management event or the (n + 6)th varbind in a third-party SNMP trap varbind.
- nnmiIncidentLifecycleStatePreviousValue, the seventh varbind (1.3.6.1.4.1.11.2.17.19.2.2.200)
- nnmiIncidentLifecycleStateCurrentValue, the eighth varbind (1.3.6.1.4.1.11.2.17.19.2.2.201)

The following table lists the possible integer values for the lifecycle states:

Name	Integer value
registered	1
inprogress	2
completed	3
closed	4
dampened	5

28.3.3 Incident correlation notifications

When the Northbound destination includes incident correlation notifications, NNMi sends incident correlation traps to the Northbound application as NNMi causal analysis correlates incidents. The Northbound application can use the information in the traps to replicate the correlation changes.

```
28. NNMi Northbound Interface
```

(1) Single correlation traps

For the single correlation trap option, the integration sends the following correlation traps:

- nnmiEvCorrelationDedup(1.3.6.1.4.1.11.2.17.19.2.0.1100)
- nnmiEvCorrelationImpact(1.3.6.1.4.1.11.2.17.19.2.0.1101)
- nnmiEvCorrelationPairwise (1.3.6.1.4.1.11.2.17.19.2.0.1102)
- nnmiEvCorrelationRate(1.3.6.1.4.1.11.2.17.19.2.0.1103)
- nnmiEvCorrelationApa(1.3.6.1.4.1.11.2.17.19.2.0.1104)
- nnmiEvCorrelationCustom(1.3.6.1.4.1.11.2.17.19.2.0.1105)

Each trap identifies one parent-child incident correlation relationship in the following varbinds:

- nnmiIncidentUuid, the sixth varbind (1.3.6.1.4.1.11.2.17.19.2.2.6)
- nnmiCorrelatedChildUuid, the seventh varbind (1.3.6.1.4.1.11.2.17.19.2.2.300)

(2) Group correlation traps

For the group correlation trap option, the integration sends the following correlation traps:

- nnmiEvCorrelationGrpDedup(1.3.6.1.4.1.11.2.17.19.2.0.2100)
- nnmiEvCorrelationGrpImpact(1.3.6.1.4.1.11.2.17.19.2.0.2101)
- nnmiEvCorrelationGrpPairwise (1.3.6.1.4.1.11.2.17.19.2.0.2102)
- nnmiEvCorrelationGrpRate (1.3.6.1.4.1.11.2.17.19.2.0.2103)
- nnmiEvCorrelationGrpApa(1.3.6.1.4.1.11.2.17.19.2.0.2104)
- nnmiEvCorrelationGrpCustom(1.3.6.1.4.1.11.2.17.19.2.0.2105)

Each trap identifies the parent-child incident correlation relationships in the following varbinds:

- nnmiIncidentUuid, the sixth varbind (1.3.6.1.4.1.11.2.17.19.2.2.6)
- nnmiCorrelatedChildrenCount, the seventh varbind (1.3.6.1.4.1.11.2.17.19.2.2.301)
- nnmiCorrelatedChildrenUuidCsv, the eighth varbind (1.3.6.1.4.1.11.2.17.19.2.2.302) This value is a comma-separated-value list of child incident UUIDs.

28.3.4 Incident deletion notifications

When the Northbound destination includes incident deletion notifications, NNMi sends an nnmiEvDeleted (1.3.6.1.4.1.11.2.17.19.2.0.3000) trap to the Northbound application when an incident is deleted in NNMi. The nnmiEvDeleted trap identifies the original incident in the sixth varbind, nnmiIncidentUuid (1.3.6.1.4.1.11.2.17.19.2.2.6).

^{28.} NNMi Northbound Interface

JP1/Network Node Manager i Setup Guide

28.3.5 Event forwarding filter

When the Northbound destination includes an incident filter, the object identifiers (OIDs) in the filter include or exclude (depending on the selected configuration option) the following event types:

- NNMi management event incidents
- Third-party SNMP traps
- nnmiEvClosed traps
- nnmiEvLifecycleStateChanged traps
- nnmiEvDeleted traps
- Correlation notification traps[#]

#: The following notes apply to correlation notification traps:

- If the incident filter prevents forwarding of the parent incident for a correlation, NNMi does not send a correlation notification trap to the Northbound application.
- If the incident filter prevents forwarding of a child incident for a correlation, the forwarded correlation notification trap does not include that child incident's UUID. In other words, if the correlation notification trap would not contain any child incident UUIDs, NNMi does not send that trap to the Northbound application.
- The DuplicateCorrelation management event is forwarded independently of the nnmiEvCorrelationDedup or nnmiEvCorrelationGrpDedup correlation notification traps. Likewise, the RateCorrelation management event is forwarded independently of the nnmiEvCorrelationRate or nnmiEvCorrelationGrpRate correlation notification traps. If the incident filter prevents forwarding of one of these correlation notification traps, NNMi might still forward the associated management events.

28.4 Changing the NNMi Northbound interface

To change the NNMi Northbound interface configuration parameters:

- 1. In the NNMi console, open the NNMi Northbound Interface Destinations form (Integration Module Configuration > Northbound Interface).
- 2. Select a destination, and then click Edit.
- 3. Modify the values as appropriate.

For details about the fields on this form, see 28.8 NNMi Northbound Interface Destination form reference.

4. Verify that the **Enabled** 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.

28.5 Disabling the NNMi Northbound interface

SNMP trap queuing does not occur while a Northbound destination is disabled.

To discontinue forwarding of NNMi incidents to a Northbound application:

- 1. In the NNMi console, open the NNMi Northbound Interface Destinations form (Integration Module Configuration > Northbound Interface).
- 2. Select a destination, and then click **Edit**. Alternatively, click **Delete** to entirely remove the configuration for the selected destination.
- 3. Clear the **Enabled** check box at the top of the form, and then click **Submit** at the bottom of the form. The changes take effect immediately.

28.6 Troubleshooting the NNMi Northbound interface

If the NNMi Northbound interface does not function as expected, follow the steps below until you have resolved the problem.

To troubleshoot the NNMi Northbound interface:

1. Verify that the trap destination port is not blocked by a firewall.

Ensure that the NNMi management server can directly address the Northbound application by host and port.

2. Verify that the integration is running correctly:

a. In the NNMi console, open the NNMi Northbound Interface Destinations form (Integration Module Configuration > Northbound Interface).

b. Select a destination, and then click **Edit**.

- c. Verify that the Enabled check box is selected.
- 3. If the Northbound destination includes management events, verify this functionality:
 - a. In the Closed Key Incidents view of the NNMi console, open any incident.

b. Set the incident lifecycle state to Registered, and then click Save.

c. Set the incident lifecycle state to Closed, and then click Save and Close.

d. After 30 seconds, determine whether the Northbound application received an nnmiEvClosed trap (or nnmiEvLifecycleStateChanged trap) for this incident.

- If the Northbound application received the trap, continue with step 4.
- If the Northbound application did not receive the trap, configure a new Northbound destination to connect with a different Northbound application, and then repeat this test from step a.

If the repeated test succeeds, the problem is with the first Northbound application. Consult that application's documentation for troubleshooting information. If the repeated test fails, contact Support for assistance.

4. If the Northbound destination includes SNMP traps, verify this functionality:

a. Generate an SNMP trap against a node in the NNMi topology by entering the following command on the NNMi management server:

```
nnmsnmpnotify.ovpl -a \
discovered node NNMi node .1.3.6.1.6.3.1.1.5.1
```

(In this command, discovered_node is the host name or IP address of a node in the NNMi topology, and NNMi_node is the host name or IP address of the NNMi management server.)

b. After 30 seconds, determine whether the Northbound application received the forwarded trap.

- If the Northbound application received the trap, the NNMi Northbound interface is working correctly.
- If the Northbound application did not receive the trap, configure a new Northbound destination to connect with a different Northbound application, and then repeat this test from step a.

If the repeated test succeeds, the problem is with the first Northbound application. Consult that application's documentation for troubleshooting information. If the repeated test fails, contact Support for assistance.

28.7 Application failover and the NNMi Northbound interface

If the NNMi management server will participate in NNMi application failover, the information in this topic applies to any integration that implements the NNMi Northbound interface for sending traps to a Northbound application.

The traps that NNMi sends to a Northbound application include the NNMi URL in the NmsUrl varbind (1.3.6.1.4.1.11.2.17.19.2.2.2). Traps received before application failover reference what is now the standby NNMi management server.

When the URL points to the standby NNMi management server, any actions that use the URL value will fail (such as launching the NNMi console).

28.7.1 Local Northbound application

If the trap-receiving component of the Northbound application is located on the NNMi management server, the following considerations apply to the configuration of the NNMi Northbound interface:

- The trap-receiving component of the Northbound application must be installed and configured identically on the active and standby NNMi management servers. Configure SNMP trap reception on the same port on both NNMi management servers.
- Configure the NNMi Northbound interface on the primary NNMi management server only.
 On the NNMi Northbound Interface Destination form, select either the NNMi FQDN or the Use Loopback option for Host identification.

At startup, the NNMi Northbound interface determines the correct name or IP address of the current NNMi management server. In this way, the Northbound interface sends traps to the trap-receiving component of the Northbound application on the active NNMi management server.

28.7.2 Remote Northbound application

If the trap-receiving component of the Northbound application is not located on the NNMi management server, configure the NNMi Northbound interface on the primary NNMi management server only. On the **NNMi Northbound Interface Destination** form, select the **Other** option for **Host** identification.

28.8 NNMi Northbound Interface Destination form reference

The **HP NNMi Northbound Interface Destination** form contains the parameters for configuring communications between NNMi and a Northbound application. This form is available from the **Integration Module Configuration** workspace. On the **NNMi Northbound Interface Destinations** form, click **New**; alternatively, select a destination, and then click **Edit**.

- Only NNMi users with the Administrator role can access the **NNMi Northbound Interface Destination** form. The **NNMi Northbound Interface Destination** form contains information for the following areas:
 - 28.8.1 NNMi Northbound application connection parameters
 - 28.8.2 NNMi Northbound interface integration content
 - 28.8.3 NNMi Northbound interface destination status information

To apply changes to the integration configuration, update the values on the **NNMi Northbound Interface Destination** form, and then click **Submit**.

28.8.1 NNMi Northbound application connection parameters

The following table lists the parameters for configuring the connection to the Northbound application.

Table 28-1: N	NMi Northbound	application	connection	information
---------------	----------------	-------------	------------	-------------

Field	Description
Host	The fully-qualified domain name (preferred) or the IP address of the server that contains the trap-receiving component of the Northbound application.
	The integration supports the following methods for identifying the server:
	• NNMi FQDN
	NNMi manages the connection to the Northbound application on the NNMi management server, and the Host field becomes read-only.
	This is the recommended configuration for Northbound applications on the NNMi management server.
	Use Loopback
	NNMi manages the connection to the Northbound application on the NNMi management server, and the Host field becomes read-only.
	• Other
	Enter a host name or IP address for identifying the Northbound application server in the Host field.
	NNMi validates that the host name or IP address in the Host field is not configured as a loopback adapter.
	This is the default configuration.
	Note: If the NNMi management server participates in NNMi application failover, see 28.7 Application failover and the NNMi Northbound interface for information about the impact of application failover on the integration.
Port	The UDP port where the Northbound application receives SNMP traps.
	Enter the port number specific to the Northbound application.
	Note: If the trap-receiving component of the Northbound application is on the NNMi management server, this port number must be different from the port NNMi uses to receive SNMP traps, as set in the SNMP Port field on the Communication Configuration form in the NNMi console.
Community String	A read-only community string for the Northbound application to receive traps.

Field	Description
Community String	If the Northbound application configuration requires a community string in the received SNMP traps, enter that value. If the Northbound application configuration does not require a specific community string, use the default value, which is public.

28.8.2 NNMi Northbound interface integration content

The following table lists the parameters for configuring the content the NNMi Northbound interface sends to the Northbound application.

	 			.	
Table 28 2.	Northhound	interface	contont	configuration	information
	NOULIDOUIU	IIIICHAUC	CONCERN	connuuration	IIIIOIIIIauoii

Field	Description
Incidents	 The incident forwarding specification: Management NNMi forwards only NNMi-generated management events to the Northbound application. 3rd Party SNMP Trap NNMi forwards only SNMP traps that NNMi receives from managed devices to the Northbound application. Syslog NNMi forwards only ArcSight Syslog messages that NNMi receives from managed devices to the Northbound application using the NorthBound Integration module. NNMi begins forwarding incidents as soon as you enable the Northbound destination. For details, see 28.3.1 Incident forwarding.
Lifecycle State Changes	 The incident change notification specification: Enhanced Closed NNMi sends an incident-closed trap to the Northbound application for each incident that changes to the CLOSED lifecycle state. This is the default configuration. State Changed NNMi sends an incident lifecycle state changed trap to the Northbound application for each incident that changes to the IN PROGESS, COMPLETED, or CLOSED lifecycle state. Both NNMi sends an incident-closed trap to the Northbound application for each incident that changes to the IN PROGESS, COMPLETED, or CLOSED lifecycle state. OMPLETED, or CLOSED lifecycle state changed trap to the Northbound application for each incident that changes to the CLOSED lifecycle state. Additionally, the integration sends an incident lifecycle state changed trap to the Northbound application for each incident that changes to the IN PROGESS, COMPLETED, or CLOSED lifecycle state. Note: In this case, each time an incident changes to the CLOSED lifecycle state, the integration sends two notification traps, an incident-closed trap and an incident lifecycle state change notifications.
Correlations	 The incident correlation notification specification: None NNMi does not notify the Northbound application of incident correlations resulting from NNMi causal analysis. This is the default configuration. Single

Field	Description
Correlations	 NNMi sends a trap for each parent-child incident correlation relationship resulting from NNMi causal analysis. Group NNMi sends one trap per correlation that lists all child incidents correlated to a parent incident. For details, see 28.3.3 Incident correlation notifications.
Deletions	 The incident deletion specification. This selection configures whether to send a deletion trap to the Northbound application for the selections made in the Incidents field: Don't Send NNMi does not notify the Northbound application when incidents are deleted in NNMi. This is the default configuration. Send NNMi sends a deletion trap to the Northbound application for each incident that is deleted in NNMi. For details, see 28.3.4 Incident deletion notifications.
NNMi Console Access	The connection protocol specification in the URL for browsing to the NNMi console from the Northbound application. The traps that NNMi sends to the Northbound application include the NNMi URL in the NmsUrl varbind (1.3.6.1.4.1.11.2.17.19.2.2.2). The configuration page defaults to the setting that matches the NNMi configuration. If the NNMi console is configured to accept both HTTP and HTTPS connections, you can change the HTTP connection protocol specification in the NNMi URL. For example, if all users of the Northbound application are on an intranet, you can set NNMi console access from the Northbound application to be over HTTP. To change the protocol for connecting to the NNMi console from the Northbound application, select the HTTP option or the HTTPS option as appropriate.
Incident Filters	 A list of object identifiers (OIDs) the integration uses to filter the events sent to the Northbound application. Each filter entry can be a valid numeric OID (such as .1.3.6.1.6.3.1.1.5.4.1.3.6.1.4.1.9) or OID prefix (such as .1.3.6.1.6.3.1.1.5.*). Select one of the following options: None NNMi sends all events to the Northbound application. This is the default configuration. Include NNMi sends only the specific events that match the OIDs identified in the filter. Exclude NNMi sends all events except for the specific events that match the OIDs identified in the filter. Exclude NNMi sends all events the text in the lower text box, and then click Add. To add a filter entry, select that entry from the list in the upper box, and then click Remove. For details, see 28.3.5 Event forwarding filter.

28.8.3 NNMi Northbound interface destination status information

The following table lists the read-only status information for the Northbound destination. This information is useful for verifying that the integration is working correctly.

Field	Description
Trap Destination IP Address	The IP address to which the destination host name resolves. This value is unique to this Northbound destination.
Uptime (seconds)	The time (in seconds) since the Northbound component was last started. The traps that NNMi sends to a Northbound application include this value in the sysUptime field (1.3.6.1.2.1.1.3.0). This value is the same for all integrations that use the NNMi Northbound interface. To see the latest value, either refresh or close then re-open the form.
NNMI URL	The URL for connecting to the NNMi console. The traps that NNMi sends to a Northbound application include this value in the NmsUrl varbind (1.3.6.1.4.1.11.2.17.19.2.2.2). This value is unique to this Northbound destination.

Table 28-3: NNMi Northbound interface destination status information

28.8.4 MIB information used by the NNMi Northbound interface

To load specific MIBs into NNMi, then view the management information used for incident notifications sent by the NNMi Northbound integration:

1. Move to the following directory:

- Windows: %NnmInstallDir%misc\nnm\snmp-mibs\Vendor\Hewlett-Packard
- Linux: /opt/OV/misc/nnm/snmp-mibs/Vendor/Hewlett-Packard
- 2. Execute the following command to load the hp-nnmi.mib file:

nnmloadmib.ovpl -load hp-nnmi.mib

3. Execute the following command to load the hp-nnmi-registrations.mib file:

nnmloadmib.ovpl -load hp-nnmi-registrations.mib

4. Execute the following command to load the hp-nnmi-nbi.mib file:

nnmloadmib.ovpl -load hp-nnmi-nbi.mib

- 5. From the NNMi console, open the Configuration workspace.
- 6. Click MIB > Loaded MIBs.
- 7. Double-click each of the MIBs you just loaded; then click MIB Variables to view the MIB information.

28.8.5 SNMP trap information used by the NNMi Northbound interface

The SNMP traps used by the Northbound interface are defined in the hp-nnmi-nbi.mib file. If you use NNMi as a Northbound application, define SNMP trap incidents.

To define SNMP trap incidents:

```
28. NNMi Northbound Interface
```

1. Perform steps 1 through 4 in 28.8.4 MIB information used by the NNMi Northbound interface.

2. Execute the following command to add the SNMP trap incident definition:

nnmincidentcfg.ovpl -loadTraps HP-NNMI-NBI-MIB

^{28.} NNMi Northbound Interface

Part 9: Integration



Integration with JP1/IM2 Intelligent Integrated Management Base

29.1 Integration with JP1/IM2 Intelligent Integrated Management Base

By linking with JP1/IM2's platform for intelligent integrated management, you can display information about nodes managed by NNMi in JP1/IM2's integrated operation viewer.

To link with JP1/IM2's platform for intelligent integrated management, you need to set up a plugin for JP1/IM2.

For details about the linkage function and the setup method, see the Release Notes.

29. Integration with JP1/IM2 Intelligent Integrated Management Base



You can use the NNMi RESTful API to link NNMi with other products.

30.1 RESTful API

You can retrieve and update the incident, node, IP address, and interface information of NNMi by using the RESTful API.

You can display and update the information of NNMi in another product or in a web portal by using the RESTful API from the other product or the web portal to link with NNMi.

For details about the RESTful API, see the Release Notes.

Appendixes

A. When NNMi Manpages Cannot Be Displayed (Linux)

If you cannot display NNMi manpages on the NNMi management server, check that the /opt/OV/man location is set in the MANPATH variable. If it is not, add the /opt/OV/man location to the MANPATH variable.

A. When NNMi Manpages Cannot Be Displayed (Linux)

B. List of MIBs Read During a New Installation

The table below lists the MIBs that are read during a new installation of NNMi.

These MIBs are not read during upgrading.

The MIB files shown in the table are relative paths from the following path:

• Windows

%NnmInstallDir%misc\nnm\snmp-mibs\

In Windows, the path demarcator is the backward slash (\setminus), not the forward slash (/).

• Linux

\$NnmInstallDir/misc/nnm/snmp-mibs/

Table B-1:	MIBs	read	during	a new	instal	lation
------------	------	------	--------	-------	--------	--------

MIB name	MIB file
ATM-FORUM-MIB	Vendor/Cisco/ATM-FORUM-MIB.my
ATM-FORUM-TC-MIB	Vendor/Cisco/ATM-FORUM-TC-MIB.my
ATM-MIB	Standard/rfc2515-ATM-MIB.mib
ATM-TC-MIB	Standard/rfc2514-ATM-TC-MIB.mib
ATM2-MIB	Standard/rfc3606-ATM2-MIB.mib
ArcsightModule	Vendor/Hewlett-Packard/hp-arcsight.mib
BGP4-MIB	Standard/rfc4273-BGP4-MIB.mib
BRIDGE-MIB	Standard/rfc4188-BRIDGE-MIB.mib
CISCO-AAL5-MIB	Vendor/Cisco/CISCO-AAL5-MIB.my
CISCO-ATM-IF-MIB	Vendor/Cisco/CISCO-ATM-IF-MIB.my
CISCO-ATM-SWITCH-ADDR-MIB	Vendor/Cisco/CISCO-ATM-SWITCH-ADDR-MIB.my
CISCO-C2900-MIB	Vendor/Cisco/CISCO-C2900-MIB.my
CISCO-CDP-MIB	Vendor/Cisco/CISCO-CDP-MIB.my
CISCO-DOT11-ASSOCIATION-MIB	Vendor/Cisco/CISCO-DOT11-ASSOCIATION-MIB.my
CISCO-DOT11-IF-MIB	Vendor/Cisco/CISCO-DOT11-IF-MIB.my
CISCO-ENTITY-FRU-CONTROL-MIB	Vendor/Cisco/CISCO-ENTITY-FRU-CONTROL-MIB.my
CISCO-ENTITY-VENDORTYPE-OID-MIB	Vendor/Cisco/CISCO-ENTITY-VENDORTYPE-OID-MIB.my
CISCO-ENVMON-MIB	Vendor/Cisco/CISCO-ENVMON-MIB.my
CISCO-FLASH-MIB	Vendor/Cisco/CISCO-FLASH-MIB.my
CISCO-FRAME-RELAY-MIB	Vendor/Cisco/CISCO-FRAME-RELAY-MIB.my
CISCO-HSRP-MIB	Vendor/Cisco/CISCO-HSRP-MIB.my
CISCO-IETF-IP-MIB	Vendor/Cisco/CISCO-IETF-IP-MIB.my
CISCO-IETF-IPMROUTE-MIB	Vendor/Cisco/CISCO-IETF-IPMROUTE-MIB.my

MIB name	MIB file
CISCO-IETF-PIM-EXT-MIB	Vendor/Cisco/CISCO-IETF-PIM-EXT-MIB.my
CISCO-IETF-PIM-MIB	Vendor/Cisco/CISCO-IETF-PIM-MIB.my
CISCO-IETF-PW-ENET-MIB	Vendor/Cisco/CISCO-IETF-PW-ENET-MIB.my
CISCO-IETF-PW-MIB	Vendor/Cisco/CISCO-IETF-PW-MIB.my
CISCO-IETF-PW-MPLS-MIB	Vendor/Cisco/CISCO-IETF-PW-MPLS-MIB.my
CISCO-IETF-PW-TC-MIB	Vendor/Cisco/CISCO-IETF-PW-TC-MIB.my
CISCO-MEMORY-POOL-MIB	Vendor/Cisco/CISCO-MEMORY-POOL-MIB.my
CISCO-MVPN-MIB	Vendor/Cisco/CISCO-MVPN-MIB.my
CISCO-NBAR-PROTOCOL-DISCOVERY- MIB	Vendor/Cisco/CISCO-NBAR-PROTOCOL-DISCOVERY-MIB.my
CISCO-PIM-MIB	Vendor/Cisco/CISCO-PIM-MIB.my
CISCO-PRODUCTS-MIB	Vendor/Cisco/CISCO-PRODUCTS-MIB.my
CISCO-QOS-PIB-MIB	Vendor/Cisco/CISCO-QOS-PIB-MIB.my
CISCO-RF-MIB	Vendor/Cisco/CISCO-RF-MIB.my
CISCO-RHINO-MIB	Vendor/Cisco/CISCO-RHINO-MIB.my
CISCO-RTTMON-MIB	Vendor/Cisco/CISCO-RTTMON-MIB.my
CISCO-RTTMON-TC-MIB	Vendor/Cisco/CISCO-RTTMON-TC-MIB.my
CISCO-SMI	Vendor/Cisco/CISCO-SMI.my
CISCO-STACK-MIB	Vendor/Cisco/CISCO-STACK-MIB.my
CISCO-TC	Vendor/Cisco/CISCO-TC.my
CISCO-VTP-MIB	Vendor/Cisco/CISCO-VTP-MIB.my
CISCOWAN-SMI	Vendor/Cisco/CISCOWAN-SMI.my
DHCP-MIB	Vendor/Microsoft/dhcp.mib
DIFFSERV-DSCP-TC	Standard/rfc3289-DIFFSERV-DSCP-TC.mib
DIFFSERV-MIB	Standard/rfc3289-DIFFSERV-MIB.mib
DISMAN-NSLOOKUP-MIB	Standard/rfc4560-DISMAN-NSLOOKUP-MIB.mib
DISMAN-PING-MIB	Standard/rfc4560-DISMAN-PING-MIB.mib
DISMAN-TRACEROUTE-MIB	Standard/rfc4560-DISMAN-TRACEROUTE-MIB.mib
DRAFT-MSDP-MIB	Vendor/Cisco/MSDP-MIB.my
DS1-MIB	Standard/rfc4805-DS1-MIB.mib
DS3-MIB	Standard/rfc3896-DS3-MIB.mib
DVMRP-MIB	Vendor/Nortel/DVMRP-MIB.mib
ENTITY-MIB	Standard/rfc4133-ENTITY-MIB.mib
ENTITY-STATE-MIB	Standard/rfc4268-ENTITY-STATE-MIB.mib
ENTITY-STATE-TC-MIB	Standard/rfc4268-ENTITY-STATE-TC-MIB.mib
EXTREME-BASE-MIB	Vendor/Extreme/v730b49.mib

MIB name	MIB file
EXTREME-CABLE-MIB	Vendor/Extreme/v730b49.mib
EXTREME-DLCS-MIB	Vendor/Extreme/v730b49.mib
EXTREME-DOS-MIB	Vendor/Extreme/v730b49.mib
EXTREME-EAPS-MIB	Vendor/Extreme/v730b49.mib
EXTREME-EDP-MIB	Vendor/Extreme/v730b49.mib
EXTREME-ENH-DOS-MIB	Vendor/Extreme/v730b49.mib
EXTREME-ESRP-MIB	Vendor/Extreme/v730b49.mib
EXTREME-FDB-MIB	Vendor/Extreme/v730b49.mib
EXTREME-FILETRANSFER-MIB	Vendor/Extreme/v730b49.mib
EXTREME-NETFLOW-MIB	Vendor/Extreme/v730b49.mib
EXTREME-NP-MIB	Vendor/Extreme/v730b49.mib
EXTREME-OSPF-MIB	Vendor/Extreme/v730b49.mib
EXTREME-PBQOS-MIB	Vendor/Extreme/v730b49.mib
EXTREME-POE-MIB	Vendor/Extreme/v730b49.mib
EXTREME-PORT-MIB	Vendor/Extreme/v730b49.mib
EXTREME-POS-MIB	Vendor/Extreme/v730b49.mib
EXTREME-QOS-MIB	Vendor/Extreme/v730b49.mib
EXTREME-RTSTATS-MIB	Vendor/Extreme/v730b49.mib
EXTREME-SERVICES-MIB	Vendor/Extreme/v730b49.mib
EXTREME-SLB-MIB	Vendor/Extreme/v730b49.mib
EXTREME-SNMPV3-MIB	Vendor/Extreme/v730b49.mib
EXTREME-STP-EXTENSIONS-MIB	Vendor/Extreme/v730b49.mib
EXTREME-SYSTEM-MIB	Vendor/Extreme/v730b49.mib
EXTREME-TRAP-MIB	Vendor/Extreme/v730b49.mib
EXTREME-TRAPPOLL-MIB	Vendor/Extreme/v730b49.mib
EXTREME-V2TRAP-MIB	Vendor/Extreme/v730b49.mib
EXTREME-VC-MIB	Vendor/Extreme/v730b49.mib
EXTREME-VLAN-MIB	Vendor/Extreme/v730b49.mib
EXTREME-WIRELESS-MIB	Vendor/Extreme/v730b49.mib
EXTREMEdot11AP-MIB	Vendor/Extreme/v730b49.mib
EXTREMEdot11f-MIB	Vendor/Extreme/v730b49.mib
EtherLike-MIB	Standard/rfc3635-EtherLike-MIB.mib
FDDI-SMT73-MIB	Standard/Historic/rfc1512-FDDI-SMT73-MIB.mib
FOUNDRY-SN-ROOT-MIB	Vendor/Foundry/FOUNDRY-SN-ROOT-MIB.mib
FRAME-RELAY-DTE-MIB	Standard/rfc2115-FRAME-RELAY-DTE-MIB.mib

B. List of MIBs Read During a New Installation
MIB name	MIB file	
FtpServer-MIB	Vendor/Microsoft/ftp.mib	
HC-RMON-MIB	Standard/rfc3273-HC-RMON-MIB.mib	
HCNUM-TC [#]	NNM/server/lib/nms-mib-model.jar/com/hp/ov/nms/mib/model/ hibernate/rfc2856-HCNUM-TC.mib	
HOST-RESOURCES-MIB	Standard/rfc2790-HOST-RESOURCES-MIB.mib	
HOST-RESOURCES-TYPES	Standard/rfc2790-HOST-RESOURCES-TYPES.mib	
HP-ICF-OID	Vendor/Hewlett-Packard/ProCurve/hpicfOid.mib	
HP-SITESCOPE-MIB	Vendor/Hewlett-Packard/HP-SITESCOPE-MIB.mib	
HP-SN-AGENT-MIB	Vendor/Hewlett-Packard/hpEtherSwitch/hp-sn-agent.mib	
HP-SN-ROOT-MIB	Vendor/Hewlett-Packard/hpEtherSwitch/hp-sn-root.mib	
HP-SN-SWITCH-GROUP-MIB	Vendor/Hewlett-Packard/hpEtherSwitch/hp-sn-switch.mib	
HP-UNIX	Vendor/Hewlett-Packard/hp-unix	
HttpServer-MIB	Vendor/Microsoft/http.mib	
IANA-ADDRESS-FAMILY-NUMBERS-MIB	Standard/IANA-ADDRESS-FAMILY-NUMBERS-MIB.mib	
IANA-MAU-MIB	Standard/rfc4836-IANA-MAU-MIB.mib	
IANA-RTPROTO-MIB	Vendor/Cisco/IANA-RTPROTO-MIB.my	
IANATn3270eTC-MIB	Standard/IANATn3270eTC-MIB.mib	
IANAifType-MIB	Standard/IANAifType-MIB.mib	
IEEE8021-TC-MIB	IEEE/IEEE8021-TC-MIB.mib	
IEEE8023-LAG-MIB	IEEE/IEEE8023-LAG-MIB.mib	
IEEE802dot11-MIB	IEEE/IEEE802dot11-MIB.mib	
IF-MIB	Standard/rfc2863-IF-MIB.mib	
IGMP-MIB	Vendor/Cisco/IGMP-MIB.my	
IGMP-STD-MIB	Vendor/Cisco/IGMP-STD-MIB.my	
INET-ADDRESS-MIB	Standard/rfc4001-INET-ADDRESS-MIB.mib	
INTEGRATED-SERVICES-MIB	Standard/rfc2213-INTEGRATED-SERVICES-MIB.mib	
IP-FORWARD-MIB	Standard/rfc4292-IP-FORWARD-MIB.mib	
IP-MIB	Standard/rfc4293-IP-MIB.mib	
IPMCAST-MIB	Standard/rfc5132-IPMCAST-MIB.mib	
IPMROUTE-MIB	Vendor/Cisco/IPMROUTE-MIB.my	
IPMROUTE-STD-MIB	Vendor/Cisco/IPMROUTE-STD-MIB.my	
IPV6-FLOW-LABEL-MIB	Standard/rfc3595-IPV6-FLOW-LABEL-MIB.mib	
IPV6-MIB	Standard/rfc2465-IPV6-MIB.mib	
IPV6-TC	Standard/rfc2465-IPV6-TC.mib	
ISDN-MIB	Standard/rfc2127-ISDN-MIB.mib	
InternetServer-MIB	Vendor/Microsoft/inetsrv.mib	

MIB name	MIB file	
JUNIPER-CHASSIS-DEFINES-MIB	Vendor/Juniper/mib-jnx-chas-defines	
JUNIPER-JS-IF-EXT-MIB	Vendor/Juniper/mib-jnx-js-if-ext	
JUNIPER-JS-SMI	Vendor/Juniper/mib-jnx-js-smi	
JUNIPER-MIB	Vendor/Juniper/mib-jnx-chassis	
JUNIPER-SMI	Vendor/Juniper/mib-jnx-smi	
JUNIPER-V1-TRAPS	Vendor/Juniper/v1_traps	
JUNIPER-VPN-MIB	Vendor/Juniper/mib-jnx-vpn	
Juniper-MIBs	Vendor/Juniper-MIBs.mib	
Juniper-UNI-SMI	Vendor/Juniper-UNI-SMI.mib	
LANGTAG-TC-MIB	Standard/rfc5131-LANGTAG-TC-MIB.mib	
LLDP-MIB	IEEE/lldp.mib	
LanMgr-Mib-II-MIB	Vendor/Microsoft/lmmib2.mib	
MAU-MIB	Standard/rfc4836-MAU-MIB.mib	
MGMD-STD-MIB	Standard/rfc5519-MGMD-STD-MIB.mib	
MPLS-L3VPN-STD-MIB	Standard/rfc4382-MPLS-L3VPN-STD-MIB.mib	
MPLS-LSR-MIB	Vendor/Cisco/MPLS-LSR-MIB.my	
MPLS-LSR-STD-MIB	Standard/rfc3813-MPLS-LSR-STD-MIB.mib	
MPLS-MIB	Vendor/Juniper/mib-jnx-mpls	
MPLS-TC-STD-MIB	Standard/rfc3811-MPLS-TC-STD-MIB.mib	
MPLS-TE-MIB	Vendor/Cisco/MPLS-TE-MIB.my	
MPLS-TE-STD-MIB	Standard/rfc3812-MPLS-TE-STD-MIB.mib	
MPLS-VPN-MIB	Vendor/Cisco/MPLS-VPN-MIB.my	
MSDP-MIB	Vendor/Nortel/MSDP-MIB.mib	
Nortel-Magellan-Passport- StandardTextualConventionsMIB	Vendor/Nortel/Nortel-Magellan-Passport- StandardTextualConventionsMIB.mib	
Nortel-Magellan-Passport- TextualConventionsMIB	Vendor/Nortel/Nortel-Magellan-Passport- TextualConventionsMIB.mib	
Nortel-Magellan-Passport- UsefulDefinitionsMIB	Vendor/Nortel/Nortel-Magellan-Passport- UsefulDefinitionsMIB.mib	
Nortel-MsCarrier-MscPassport- StandardTextualConventionsMIB	Vendor/Nortel/Nortel-MsCarrier-MscPassport- StandardTextualConventionsMIB.mib	
Nortel-MsCarrier-MscPassport- TextualConventionsMIB	Vendor/Nortel/Nortel-MsCarrier-MscPassport- TextualConventionsMIB.mib	
Nortel-MsCarrier-MscPassport- UsefulDefinitionsMIB	Vendor/Nortel/Nortel-MsCarrier-MscPassport- UsefulDefinitionsMIB.mib	
OLD-CISCO-CHASSIS-MIB	Vendor/Cisco/OLD-CISCO-CHASSIS-MIB.my	
OLD-CISCO-INTERFACES-MIB	Vendor/Cisco/OLD-CISCO-INTERFACES-MIB.my	
OLD-CISCO-SYS-MIB	Vendor/Cisco/OLD-CISCO-SYS-MIB.my	

MIB name	MIB file	
OSPF-MIB	Standard/rfc4750-OSPF-MIB.mib	
P-BRIDGE-MIB	Standard/rfc4363-P-BRIDGE-MIB.mib	
PIM-MIB	Vendor/Cisco/PIM-MIB.my	
PIM-STD-MIB	Standard/rfc5060-PIM-STD-MIB.mib	
POWER-ETHERNET-MIB	Standard/rfc3621-POWER-ETHERNET-MIB.mib	
PerfHist-TC-MIB	Standard/rfc3593-PerfHist-TC-MIB.mib	
Q-BRIDGE-MIB	Standard/rfc4363-Q-BRIDGE-MIB.mib	
RAPID-CITY	Vendor/Nortel/RAPID-CITY.mib	
RFC-1212 [#]	NNM/server/lib/nms-mib-model.jar/com/hp/ov/nms/mib/model/ hibernate/rfc1212-RFC1212.mib	
RFC-1215	Standard/rfc1215-RFC1215.mib	
RFC1155-SMI [#]	NNM/server/lib/nms-mib-model.jar/com/hp/ov/nms/mib/model/ hibernate/rfc1155-RFC1155-SMI.mib	
RFC1213-MIB [#]	NNM/server/lib/nms-mib-model.jar/com/hp/ov/nms/mib/model/ hibernate/rfc1213-RFC1213-MIB.mib	
RFC1271-MIB [#]	NNM/server/lib/nms-mib-model.jar/com/hp/ov/nms/mib/model/ hibernate/rfc1271-RFC1271-MIB.mib	
RFC1315-MIB	Standard/rfc1315-RFC1315-MIB.mib	
RIPv2-MIB	Standard/rfc1724-RIPv2-MIB.mib	
RMON-MIB	Standard/rfc2819-RMON-MIB.mib	
RMON2-MIB	Standard/rfc4502-RMON2-MIB.mib	
RS-232-MIB	Standard/rfc1659-RS-232-MIB.mib	
SMON-MIB	Standard/rfc2613-SMON-MIB.mib	
SNMP-FRAMEWORK-MIB	Standard/rfc3411-SNMP-FRAMEWORK-MIB.mib	
SNMP-REPEATER-MIB	Standard/rfc2108-SNMP-REPEATER-MIB.mib	
SNMP-TARGET-MIB	Standard/rfc3413-SNMP-TARGET-MIB.mib	
SNMP-VIEW-BASED-ACM-MIB	Standard/rfc3415-SNMP-VIEW-BASED-ACM-MIB.mib	
SNMPv2-CONF [#]	NNM/server/lib/nms-mib-model.jar/com/hp/ov/nms/mib/model/ hibernate/rfc1904-SNMPv2-CONF.mib	
SNMPv2-MIB [#]	NNM/server/lib/nms-mib-model.jar/com/hp/ov/nms/mib/model/ hibernate/rfc3418-SNMPv2-MIB.mib	
SNMPv2-SMI [#]	NNM/server/lib/nms-mib-model.jar/com/hp/ov/nms/mib/model/ hibernate/rfc2578-SNMPv2-SMI.mib	
SNMPv2-TC [#]	NNM/server/lib/nms-mib-model.jar/com/hp/ov/nms/mib/model/ hibernate/rfc2579-SNMPv2-TC.mib	
SONET-MIB	Standard/rfc3592-SONET-MIB.mib	
TOKEN-RING-RMON-MIB	Standard/Historic/rfc1513-TOKEN-RING-RMON-MIB.mib	
TRANSPORT-ADDRESS-MIB	Standard/rfc3419-TRANSPORT-ADDRESS-MIB.mib	
TUNNEL-MIB	Standard/rfc4087-TUNNEL-MIB.mib	

MIB name	MIB file	
VMWARE-AGENTCAP-MIB	Vendor/VMware/VMWARE-AGENTCAP-MIB.mib	
VMWARE-ENV-MIB	Vendor/VMware/VMWARE-ENV-MIB.mib	
VMWARE-OBSOLETE-MIB	Vendor/VMware/VMWARE-OBSOLETE-MIB.mib	
VMWARE-PRODUCTS-MIB	Vendor/VMware/VMWARE-PRODUCTS-MIB.mib	
VMWARE-RESOURCES-MIB	Vendor/VMware/VMWARE-RESOURCES-MIB.mib	
VMWARE-ROOT-MIB	Vendor/VMware/VMWARE-ROOT-MIB.mib	
VMWARE-SYSTEM-MIB	Vendor/VMware/VMWARE-SYSTEM-MIB.mib	
VMWARE-TC-MIB	Vendor/VMware/VMWARE-TC-MIB.mib	
VMWARE-VC-EVENT-MIB	ENT-MIB Vendor/VMware/VMWARE-VC-EVENT-MIB.mib	
VMWARE-VMINFO-MIB	Vendor/VMware/VMWARE-VMINFO-MIB.mib	
VPN-TC-STD-MIB)-MIB Standard/rfc4265-VPN-TC-STD-MIB.mib	
VRRP-MIB	Standard/rfc2787-VRRP-MIB.mib	
WINDOWS-NT-PERFORMANCE	RMANCE Vendor/Microsoft/WINDOWS-NT-PERFORMANCE.mib	
WINS-MIB	Vendor/Microsoft/wins.mib	
X-DDI-MIB	Vendor/Nortel/x-ddi-adapter-mib	
XYLAN-BASE-MIB	Vendor/OTHER-VENDORS/XYLAN-BASE-MIB.mib	
XYLAN-HEALTH-MIB	Vendor/OTHER-VENDORS/XYLAN-HEALTH-MIB.mib	

#: Some MIB files are contained in JAR files. The MIB files in the table are relative paths within the JAR files shown below.

Windows

%NnmInstallDir%NNM\server\lib\nms-mib-model.jar

Linux

\$NnmInstallDir/NNM/server/lib/nms-mib-model.jar

B. List of MIBs Read During a New Installation

NNMi provides many environment variables that are available for your use in navigating the file system and writing scripts.

C.1 Environment variables used in this manual

This manual primarily uses the following two NNMi environment variables to reference file and directory locations. This list shows the default values. The actual values depend on the selections that you made during NNMi installation.

• Windows:

```
- %NnmInstallDir%: drive: \Program Files (x86) \Hitachi\Cm2NNMi\
```

```
- %NnmDataDir%: drive: \ProgramData\Hitachi\Cm2NNMi\
```



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

Important

If a path name contains a space, enclose the entire path name in double quotation marks (").

Example:

```
"%NnmInstallDir%bin\ovstatus"-c
```

- Linux:
 - \$NnmInstallDir:/opt/OV

```
- $NnmDataDir: /var/opt/OV
```

Note

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

The following environment variables are should be replaced with the following path in this manual.

• Windows:

```
%jdkdir%: %NnmInstallDir%nonOV\jdk\zulu\zulu8.21.0.1-jdk8.0.131-win_x64
```

```
• Linux:
$jdkdir: /opt/OV/nonOV/jdk/zulu/zulu8.21.0.1-jdk8.0.131-linux_x64
```

Additionally, this document references some of the NNMi environment variables that you can source as part of your user log-on configuration on the NNMi management server. These variables are of the form NNM_*. For details about this extended list of NNMi environment variables, see C.2 Other available environment variables.

C. NNMi Environment Variables

C.2 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.

To set up the extended list of NNMi environment variables, use a command similar to the following example:

- Windows: C:\Program Files (x86)\Hitachi\Cm2NNMi\bin\nnm.envvars.bat
- Linux: . /opt/OV/bin/nnm.envvars.sh A space must be inserted between the period (.) and the forward slash (/).

After you run the command for your operating system, you can use the NNMi environment variables shown in Table C-1 (Windows) or Table C-2 (Linux) to get to commonly used NNMi file locations.

Table C-1: Environment variable default locations for the Windows operating system

Variable	Windows (example)	
%NNM_BIN%	C:\Program Files (x86)\Hitachi\Cm2NNMi\bin	
%NNM_CONF%	C:\ProgramData\Hitachi\Cm2NNMi\Conf	
%NNM_DATA%	C:\ProgramData\Hitachi\Cm2NNMi	
%NNM_DB%	C:\ProgramData\Hitachi\Cm2NNMi\shared\nnm\databases	
%NNM_JAVA%	C:\Program Files (x86)\Hitachi\Cm2NNMi\nonOV\jdk\hpsw\bin \java.exe	
%NNM_JAVA_DIR%	C:\Program Files (x86)\Hitachi\Cm2NNMi\java	
%NNM_JBOSS%	C:\Program Files (x86)\Hitachi\Cm2NNMi\nmsas	
%NNM_JBOSS_DEPLOY%	C:\Program Files (x86)\Hitachi\Cm2NNMi\nmsas\server\nms\deploy	
%NNM_JBOSS_LOG%	C:\ProgramData\Hitachi\Cm2NNMi\log\nnm	
%NNM_JBOSS_SERVERCONF%	C:\Program Files (x86)\Hitachi\Cm2NNMi\nmsas\server\nms	
%NNM_JRE%	C:\Program Files (x86)\Hitachi\Cm2NNMi\nonOV\jdk\hpsw	
%NNM_LOG%	C:\ProgramData\Hitachi\Cm2NNMi\log	
%NNM_LRF%	C:\ProgramData\Hitachi\Cm2NNMi\shared\nnm\lrf	
%NNM_PRIV_LOG%	C:\ProgramData\Hitachi\Cm2NNMi\log	
%NNM_PROPS%	C:\ProgramData\Hitachi\Cm2NNMi\shared\nnm\conf\props	
%NNM_SHARED_CONF%	C:\ProgramData\Hitachi\Cm2NNMi\shared\nnm\conf	
%NNM_SHARE_LOG%	C:\ProgramData\Hitachi\Cm2NNMi\log	
%NNM_SNMP_MIBS%	C:\Program Files (x86)\Hitachi\Cm2NNMi\misc\nnm\snmp-mibs	
%NNM_TMP%	C:\ProgramData\Hitachi\Cm2NNMi\tmp	
%NNM_USER_SNMP_MIBS%	C:\ProgramData\Hitachi\Cm2NNMi\shared\nnm\user-snmp-mibs	
%NNM_WWW%	C:\ProgramData\Hitachi\Cm2NNMi\shared\nnm\www	

Table C-2: Environment variable default locations for the Linux operating system

Variable	Linux (example)	
\$NNM_BIN	/opt/OV/bin	
\$NNM_CONF	/var/opt/OV/conf	
\$NNM_DATA	/var/opt/OV	
\$NNM_DB	/var/opt/OV/shared/nnm/databases	
\$NNM_JAVA	/opt/0V/nonOV/jdk/hpsw/bin/java	
\$NNM_JAVA_DIR	/opt/0V/java	
\$NNM_JBOSS	/opt/OV/nmsas	
\$NNM_JBOSS_DEPLOY	/opt/OV/nmsas/server/nms/deploy	
\$NNM_JBOSS_LOG	/var/opt/OV/log/nnm	
\$NNM_JBOSS_SERVERCONF	/opt/OV/nmsas/server/nms	
\$NNM_JRE	/opt/0V/nonOV/jdk/hpsw	
\$NNM_LOG	/var/opt/OV/log	
\$NNM_LRF	/var/opt/OV/shared/nnm/lrf	
\$NNM_PRIV_LOG	/var/opt/OV/log	
\$NNM_PROPS	/var/opt/OV/shared/nnm/conf/props	
\$NNM_SHARED_CONF	/var/opt/OV/shared/nnm/conf	
\$NNM_SHARE_LOG	/var/opt/OV/log	
\$NNM_SNMP_MIBS	/opt/OV/misc/nnm/snmp-mibs	
\$NNM_USER_SNMP_MIBS	/var/opt/OV/shared/nnm/user-snmp-mibs	
\$NNM_TMP	/var/opt/OV/tmp	
\$NNM_WWW	/var/opt/OV/shared/nnm/www	

D. The Causal Engine and NNMi Incidents

Communications and data networks have grown significantly in size and complexity, and so has the number of faults that occur. A single failure can trigger many alarms. Distinguishing real problems from anecdotal alarms has become a bottleneck for network operators. Traditional event correlation systems are able to reduce the number of alarms, but these systems tend to fall short in terms of identifying the root causes in an automated manner.

The NNMi Causal Engine technology applies *root cause analysis* (RCA) to network symptoms, using a causality-based approach to incident generation.

D.1 Causal relationship analysis - advanced consideration

Causal Engine technology provides the following advanced features:

- Uses the NmsApa (NMS Active Problem Analyzer) JBoss service to analyze your network
- Uses a 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.
 - Effectively handles ambiguity and partial symptoms.
- Is dynamic
 - Actively solicits symptoms during analysis.
 - Reacts dynamically to topology changes.
- Is extendable
 - 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.

D.2 Causal Engine concept

Causal Engine uses the following sequential approach to incident generation:

- 1. Formally define the root-cause problems and symptoms.
- 2. Perform analysis by relating symptoms to root-cause problems using the behavioral and object models. Symptoms come from two sources:
 - State Poller, where the symptoms are state changes
 - Events, where the symptoms are traps
- 3. Generate conclusions that relate to the root causes.

Causal Engine conclusions include the artifacts related to the model, which include the following details:

- Generated incidents
- Correlated incidents
- Suppressed incidents
- Canceled incidents

D. The Causal Engine and NNMi Incidents

• Status on relevant objects

D.3 Concept of status

In addition to incident manipulation, the NmsApa service sets status on relevant objects. Status indicates the overall health of an object and is determined from the outstanding conclusions. Every conclusion has a severity associated with it. The status reported is the most severe of all outstanding conclusions. In addition, conclusions inform the user of the root cause (or reason) for an object's status.

The NmsApa service manages the following objects:

- SNMP agents
- IPv4 addresses
- Interfaces
- Connections
- Nodes
- Node groups

The NmsApa service uses the following status categories in decreasing order of severity:

- Unknown
- Disabled
- Critical
- Major
- Minor
- Warning
- Normal
- No Status

D.4 About episodes

The goal of the NmsApa service is to present a single incident that the operator or network engineer can investigate. 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 the incident configuration.

Examples

- The AddressNotResponding incident is suppressed by the InterfaceDown incident, according to the following scenario:
 - When an IP address stops responding to ICMP, an episode begins, which exists for a duration of 60 seconds.
 - Within that duration, if the interface associated with that IP address goes down, the NmsApa service concludes that an interface down condition caused the IPv4 address to stop responding.
 - Therefore, the AddressNotResponding incident is suppressed.

D. The Causal Engine and NNMi Incidents

Only the InterfaceDown incident is generated.

- To ensure that the InterfaceDown incident is discovered within that period, the NmsApa service issues specified polling to that interface. This enables the network engineer to correct the root cause of the problem (the interface in this case).
- 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, NNMi generates an InterfaceDown incident. In this case, the network engineer must 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 begins for the neighboring node, which exists for a duration of 300 seconds.
 - Within that duration, if the node goes down, NNMi correlates the InterfaceDown incident 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.

D.5 What does NNMi analyze?

Using SNMP, NNMi utilizes SNMP agents (processes running on managed nodes that provide management functions) to acquire information from managed nodes. The SNMP agents manage the interfaces and ports on the managed nodes, and can associate them with one or more nodes.

The following list shows the status categories that can be associated with SNMP agents:

- Unknown Not applicable
- Disabled Not applicable
- Critical Indicates the SNMP agent does not respond to SNMP queries.
- Minor Not applicable
- Warning Not applicable
- Normal Indicates the SNMP agent responds to SNMP queries.
- No Status Indicates the SNMP agent is not polled.

An *IPv4 address* is a routable address that responds to ICMP. An IPv4 address is normally associated with a node. NNMi reports the status of a node as follows:

- Unknown Not applicable
- Disabled Indicates the interface associated with this IPv4 address cannot be managed or is disabled.
- Critical Indicates the IPv4 address does not respond to ICMP queries (pings the device).
- Minor Not applicable
- Warning Not applicable
- Normal Indicates the IPv4 address responds to ICMP queries.
- No Status Indicates the IPv4 address is not polled.

D. The Causal Engine and NNMi Incidents

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 does not respond to SNMP queries. The NmsApa service cannot determine the health because the ifAdminStatus and ifOperStatus values cannot be measured.
- Disabled Indicates the interface is administratively down (ifAdminStatus=down).
- Critical Indicates the interface is operationally down (ifOperStatus=down).
- Minor Not applicable
- Warning Not applicable
- Normal Indicates the interface is operationally up (ifOperStatus=up).
- No Status Indicates the 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 the following two categories:

- 1. Network nodes, which are active devices such as switches, routers, bridges, and hubs
- 2. End nodes, such as Linux or Windows servers

NNMi typically manages network nodes, reporting node status as follows:

- Unknown The SNMP agent associated with the node does not respond to SNMP queries and the polled IPv4 address does not respond to the ICMP query. This indicates that NNMi is unable to manage the node.
- Disabled Not applicable
- Critical Indicates any one of the following:
 - The node is down as determined by neighbor analysis.
 - The node is marked as important and is unmanageable (NNMi cannot access the node from the NNMi server).
 - The node is an island (if it has no neighbor), and therefore is unmanageable.
 - The NmsApa service cannot determine whether the node is down or the incoming connection is down.
- Minor Indicates any one of the following:
 - The SNMP agent associated with the node does not respond to SNMP queries.
 - At least one interface in the node is down.
 - At least one IPv4 address on the node does not respond to ICMP.
- Warning Not applicable
- Normal Indicates the SNMP agent of the node, polled interfaces, and polled IPv4 addresses are up.
- No Status Indicates 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 *Cisco Discovery Protocol* (CDP) and *Extreme Discovery Protocol* (EDP). NNMi reports the status of a connection as follows:

- Unknown Indicates all endpoints of the connection have Unknown status.
- Disabled Indicates one endpoint of the connection is disabled.
- Critical Indicates all endpoints are operationally down.
- Minor Indicates one endpoint is down.

D. The Causal Engine and NNMi Incidents

- Warning Indicates endpoints have unknown and non-critical status.
- Normal Indicates all endpoints are operationally up.
- No Status Indicates one endpoint is not polled.

A *node group* is a logical collection of nodes created by an NNMi administrator to customize the polling configuration. For example, some nodes, such as routers, might be critical to a business and therefore must be polled more frequently. For such a case, the NNMi administrator would 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 Indicates all nodes in the group have Unknown status.
- Disabled Not applicable
- Critical Indicates all nodes in the group have Critical status.
- Minor Indicates at least one node in the group has Critical status.
- Warning Indicates nodes have unknown and non-critical status.
- Normal Indicates all nodes in the group have Normal status.
- No Status Indicates all nodes in the group have no status.

D.6 Failure scenarios

The following subsections describe the network fault scenarios that the NNMi Causal Engine analyzes and how the failures are diagnosed. The following table shows incident examples indicated by these scenarios, together with other examples:

Incident name	Description	
AddressNotResponding	IPv4 address does not respond to ICMP. The following are possible reasons:1. The node is down.2. Because of an error in the device (such as a router) configuration, several IPv4 addresses cannot be reached.	
InterfaceDown	The interface is not up.	
ConnectionDown	Both (or all) connection endpoints are down.	
NodeDown	 This incident indicates that the NmsApa service has determined that the node is down based on the following analysis: 100% of the IPv4 addresses assigned to this node cannot be reached. The SNMP agents installed in this machine are not responding. Indicates that at least two neighboring devices can be reached, but reports that there is a problem in the connectivity to this node. 	
NodeOrConnectionDown	This incident indicates that the node is not responding to ICMP or SNMP queries. Moreover, because only one of the neighboring interfaces is down, the NmsApa service cannot determine whether the node is down or the connection is down.	

(1) 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 example, the community string for this SNMP agent has been changed, or NNMi's communication configuration settings have not yet been updated, but the node is operational (IPv4 address can be pinged).

Root cause: The SNMP agent is not responding.

Incident: An SNMPAgentNotResponding incident is generated.

Status: The SNMP agent is in Critical status.

Conclusion: SNMPAgentNotResponding

Effect: The node status is Minor. The 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.

(2) 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: This scenario continues the previous (1) SNMP agent not responding to SNMP queries scenario. An NNMi administrator has updated the community 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. The SNMPAgentNotResponding incident is closed.

Status: SNMP agent is in Normal state.

Conclusion: SNMPAgentResponding

D. The Causal Engine and NNMi Incidents

Effect: The node status is Normal. The conclusion on the node is ResponsiveAgentInNode. InterfaceUnmanageable is cleared from all polled interfaces and the interfaces return to their previous status.

(3) IPv4 address not responding to ICMP



Scenario: IPv4 address 1 on Server 1 (S1) is not responding. For example, the route on Router 1 (R1) has changed from Interface 1 to Interface 2, so that packets destined for Interface 1 on Server 1 are now routed out of Interface 2 on Router 1. 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: An AddressNotResponding incident is generated.

Status: IPv4 address is in Critical status.

Conclusion: AddressNotResponding

Effect: The node status is Minor. The conclusion on the node is SomeUnresponsiveAddressesInNode.

(4) IPv4 address responding to ICMP



Scenario: This scenario continues the previous (3) IPv4 address not responding to ICMP scenario. 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, or the SNMP agent is up, or both).

Root cause: IPv4 address is responding.

Incident: None generated. The AddressNotResponding incident is closed.

Status: The IPv4 address is in Normal state.

Conclusion: AddressResponding

Effect: The node status is Normal. The conclusion on the node is ResponsiveAddressesInNode.

(5) Interface is operationally down



Scenario: Interface 1 for R1 is operationally down (ifOperStatus=down) and administratively up (ifAdminStatus=up). Router 1 sends a LinkDown trap. Router 1 can be reached because some IPv4 addresses, such as IPv4 address 2, respond to ping. The SNMP agent is up. IPv4 address 1 is associated with Interface 1 and has stopped responding to ICMP.

Root cause: The interface is down.

Incident: An InterfaceDown incident is generated. The LinkDown incident is correlated beneath the InterfaceDown incident.

Status: The interface is in Critical status.

Conclusion: InterfaceDown

Effect: The node status is Minor. The conclusion on the node is InterfacesDownInNode. No AddressNotResponding incident is associated with the IPv4 address.

D. The Causal Engine and NNMi Incidents

(6) Interface is operationally up



Scenario: This scenario continues the previous (5) Interface is operationally down scenario. Interface 1 for R1 is now operationally up (ifOperStatus=up). The node can be reached. All of its IPv4 addresses respond to ping. The SNMP agent is up.

Root cause: The interface is up.

Incident: None generated. The InterfaceDown incident is closed.

Status: The interface is in Normal status.

Conclusion: InterfaceUp

Effect: The node status is Normal. The conclusion on the node is InterfacesUpInNode.

(7) Interface is administratively down



Scenario: Interface 1 for R1 is administratively down (ifAdminStatus=down), but the node can be reached. For example, Interface 2 responds to ping and the SNMP agent is up. Disabling Interface 1 for R1 brings that interface operationally down. The IPv4 address associated with this interface, IPv4 address 1, stops responding to ICMP.

Root cause: Interface 1 for R1 is disabled.

Incident: None generated.

D. The Causal Engine and NNMi Incidents

Status: The interface is in Disabled status.

Conclusion: InterfaceDisabled

Effect: The IPv4 address associated with Interface 1 for R1 has a status of Disabled. The conclusion on the IPv4 address is AddressDisabled.

(8) Interface is administratively up



Scenario: This scenario continues the previous (5) Interface is operationally down scenario. Interface 1 for R1 is now administratively up (ifAdminStatus=up). The node can be reached because some of the IPv4 addresses of that interface respond to ping. The SNMP agent is up. Enabling Interface 1 for R1 brings it operationally up. The IPv4 address associated with this interface starts responding to ICMP.

Root cause: The interface is enabled.

Incident: None generated.

Status: The interface is in Normal status.

Conclusion: InterfaceEnabled

Effect: The IPv4 address associated with Interface 1 for R1 has a status of Enabled. The conclusion on the IPv4 address is AddressEnabled.

(9) Connection is operationally down



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: A ConnectionDown incident is generated. The InterfaceDown incident from If13 and If31 are correlated beneath ConnectionDown.

Status: The connection is in Critical status.

Conclusion: ConnectionDown

(10) Connection is operationally up



Scenario: This scenario continues the previous (9) Connection is operationally down scenario. The connection between If13 and If31 is now up.

Root cause: The connection between If13 and If31 is up.

Incident: None generated. The ConnectionDown incident is closed.

D. The Causal Engine and NNMi Incidents

Status: The connection is in Normal status.

Conclusion: ConnectionUp





Scenario: Access switches ASW11, ASW12, ASW21, and ASW22 are redundantly connected to the distribution routers as shown. Distribution routers DR1 and DR2 are directly connected to each another. Distribution router DR1 goes down.

Root cause: Node DR1 is down according to neighbor analysis.

Incident: A NodeDown incident is generated. The InterfaceDown incidents from one-hop neighbors are correlated beneath the NodeDown incident.

Status: The node is in Critical status.

Conclusion: NodeDown

(12) Directly connected node is up



Scenario: This scenario continues the previous (11) Directly connected node is down scenario. Distribution router DR1 comes back up.

Root cause: Node DR1 is up.

Incident: None generated. The NodeDown incident is closed.

Status: The node is in Normal status.

Conclusion: NodeUp

(13) Indirectly connected node is down





The diagram 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 through optical repeaters (because the optical repeaters do not respond to SNMP or ICMP queries, they are not discovered by NNMi).

D. The Causal Engine and NNMi Incidents

Router 2 becomes unreachable, either because its connected interface is down or because the connection between the optical repeaters is down. The interface on Router 1 that indirectly connects it to Router 2 is still up because its optical repeater is still up.

Root cause: Router 2 is down according to neighbor analysis.

Incident: A NodeDown incident is generated.

Status: Node router R2 is in Critical status.

Conclusion: NodeDown

(14) Indirectly connected node is up



Legend: R1: Router 1 OR: Optical repeater (not discovered by NNMi) R2: Router 2

Note

The diagram is conceptual. It does not represent an actual NNMi topology map or workspace view.

Scenario: This scenario continues the previous (13) Indirectly connected node is down scenario. The failed connection comes back up. Router 2 becomes reachable.

Root cause: The connection between Router 1 and Router 2 is up.

Incident: None generated. The NodeDown incident is closed.

Status: Router 2's status is Normal. The connection Status is Normal.

Conclusion: NodeUp

D. The Causal Engine and NNMi Incidents

(15) Directly connected node is down and creates a shadow



Scenario: Router 2 (R2) goes down as shown above.

Root cause: Node (Router 2) is down according to NNMi's neighbor analysis.

Incident: A NodeDown incident is generated. The InterfaceDown incidents from one-hop neighbors are correlated beneath the NodeDown incident.

Status: The 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.

D. The Causal Engine and NNMi Incidents

(16) Directly connected node is up, clearing the shadow



ASW2: Access switch 2 ASW3: Access switch 3

ASW4: Access switch 4

Scenario: This scenario continues the previous (15) Directly connected node is down and creates a shadow scenario. Router 2 comes back up.

Root cause: Node Router 2 is up.

Incident: None generated. The NodeDown incident is closed by the NodeUp incident.

Status: The 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.

(17) Important node is unreachable

Scenario: A node that is part of the Important Nodes node group cannot be reached.

Note

You must add a node to the Important Nodes node group before the NmsApa service can analyze it. 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: The node is down. The NmsApa service does not do neighbor analysis, but concludes that the node is down because it was marked as important.

D. The Causal Engine and NNMi Incidents

Incident: A NodeDown incident is generated. There are no correlated incidents.

Status: The node is in Critical status.

Conclusion: NodeDown

(18) Important node is reachable

Scenario: This scenario continues the previous (17) Important node is unreachable scenario. The important node comes back up and can be reached.

Root cause: The node is up.

Incident: None generated. The NodeDown incident is closed by the NodeUp incident.

Status: The node is in Normal status.

Conclusion: NodeUp

(19) Node or connection is down



Scenario: There is no redundancy for Router 2 (R2). Either Router 2 is down or the connection between Router 1 (R1) and Router 2 is down.

Root cause: The node or the connection is down.

Incident: The NodeOrConnectionDown incident is generated. The source node in this scenario is Router 2.

Status: The Node is in Critical status. The connection is in Minor status.

Conclusion: NodeOrConnectionDown

(20) Node or connection is up



Scenario: This scenario continues the previous (19) Node or connection is down scenario. Router 2 is now up.

D. The Causal Engine and NNMi Incidents

Root cause: NodeUp

Incident: None generated. The NodeOrConnectionDown incident is closed.

Status: The node is in Normal status. The connection is in Normal status.

Conclusion: NodeUp

(21) Island group is down



Note

Ë

The diagram 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.

Note

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 can 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. For details about island groups, see *NNMi Console* in NNMi Help.

Root cause: Island Group 2 is down according to neighbor analysis.

Incident: The IslandGroupDown incident is generated. NNMi chooses a representative node from Island Group 2 as the source node for the incident.

Status: The status of Island Group 2 is set to Unknown. Objects in Island Group 2 have Unknown status. The connecting interface from Island Group 1 is up because the connection from the interface to the WAN is still up.

Conclusion: Not applicable for island groups.

(22) Island group is up



Note

The diagram is conceptual. It does not represent an actual NNMi topology map or workspace view.

Scenario: This scenario continues the previous (21) Island group is down scenario. The service provider's WAN problems are fixed, and Island Group 2 can be reached.

Root cause: The WAN connection to Island Group 2 is back up.

Incident: None generated. The IslandGroupDown incident is closed.

Status: The status for Island Group 2 is set to Normal. Objects in Island Group 2 return to Normal status.

Conclusion: Not applicable for island groups.

(23) Link aggregated ports (NNMi Advanced)

Aggregator is up



• If12, If22

• If13, If23

Scenario: All ports within the port aggregator are operationally and administratively up.

Root cause: All operational and administrative states are up.

Incident: No incident is generated.

Status: The status of the aggregator is set to Normal.

D. The Causal Engine and NNMi Incidents

Aggregator is degraded



IIII, II2I
If12, If22

If12, II22
If13, If23

• 1113, 1123

Scenario: Some (but not all) ports within the port aggregator are operationally down.

Root cause: Operational states on some ports are down.

Incident: An AggregatorDegraded incident is generated.

Status: The status of the aggregator is set to Minor.

Conclusion: AggregatorDegraded

Aggregator is down



- If11, If21
- If12, If22
 If13 If23
- If13, If23

Scenario: All ports within the port aggregator are operationally down.

Root cause: Operational states on all ports are down.

Incident: An AggregatorDown incident is generated.

Status: The status of the aggregator is set to Critical.

Conclusion: AggregatorDown

D. The Causal Engine and NNMi Incidents

(24) Link aggregated connections (NNMi Advanced)



Link aggregated connection is up

If11, If21

• If12, If22

• If13, If23

Scenario: All port aggregator members of the connection are up.

Root cause: The aggregator is up on all members of the connection.

Incident: No incident is generated.

Status: The status of the aggregated connection is set to Normal.

Conclusion: AggregatorLinkUp

Link aggregated connection is degraded



• If11. If21

- If12, If22 ٠
- If13, If23 •

Scenario: Some (but not all) port aggregator members of the connection are down.

Root cause: The aggregator is down on some members of the connection.

Incident: An AggregatorLinkDegraded incident is generated.

Status: The status of the aggregated connection is set to Minor.

Conclusion: AggregatorLinkDegraded

D. The Causal Engine and NNMi Incidents

Link aggregated connection is down



- If11, If21
 If12 If22
- If12, If22
 If12, If22
- If13, If23

Scenario: All port aggregator members of the connection are down.

Root cause: The aggregator is down on all members of the connection.

Incident: An AggregatorLinkDown incident is generated.

Status: The status of the aggregated connection is set to Critical.

Conclusion: AggregatorLinkDown

(25) Router redundancy groups: HSRP and VRRP (NNMi Advanced)

Router redundancy group has no primary



Scenario: A router redundancy group does not have a primary member. A properly functioning HSRP or VRRP router redundancy group must have one operational primary router and one operational secondary router.

Root cause: This scenario could be the result of an interface on the primary router failing, when the secondary was not active or there was a misconfiguration of the router redundancy group.

D. The Causal Engine and NNMi Incidents

Incident: An RrgNoPrimary incident is generated. The RrgNoPrimary incident is impacted. If there is an identified root cause such as InterfaceDown, the InterfaceDown incident is correlated under the RrgNoPrimary incident as an impact correlation.

Status: The status of the router redundancy group is set to Critical.

Conclusion: RrgNoPrimary



Router redundancy group has multiple primaries

Scenario: A router redundancy group has multiple routers reporting as the primary router. A properly functioning HSRP or VRRP router redundancy group must have only one operational primary router.

Root cause: This scenario could be due to a faulty configuration of the router redundancy group.

Incident: An RrgMultiplePrimary incident is generated. The RrgMultiplePrimary incident is impacted.

Status: The status of the router redundancy group is set to Major.

Conclusion: RrgMultiplePrimary

Router redundancy group has failed over



JP1/Network Node Manager i Setup Guide

Scenario: A router redundancy group has had a failure on the primary router and the secondary router has taken over as primary. The standby usually becomes the secondary, which is not a problem; the group is functioning as intended. The incident generated for this scenario is for informational purposes to report that the group has had a failover.

Root Cause: This scenario is most likely due to a failure on the primary router.

Incident: An RrgFailover incident is generated. The correlation nature of RrgFailover is service impact. If an identified root cause such as InterfaceDown exists, the InterfaceDown incident is correlated under the RrgFailover incident as an impact correlation.

Status: None generated.

Conclusion: RrgFailover

Router redundancy group has no secondary



Scenario: A router redundancy group has had a failure on the secondary router. Either there is no standby or the standby did not take over as the secondary.

Root Cause: This scenario could be due to an interface failure on the router or some misconfiguration of the router redundancy group.

Incident: An RrgNoSecondary incident is generated. The correlation nature of RrgNoSecondary is service impact. If an identified root cause such as InterfaceDown exists, the correlation nature between the RrgNoSecondary and InterfaceDown interfaces is service impact.

Status: The status of the router redundancy group is set to Minor.

Conclusion: RrgNoSecondary

D. The Causal Engine and NNMi Incidents

Router redundancy group has multiple secondaries



R3: Standby Router 3, acting as secondary

Scenario: A router redundancy group has multiple routers reporting as the secondary router. A properly functioning HSRP or VRRP router redundancy group must have only one operational secondary router.

Root Cause: This scenario could be due to misconfiguration of the router redundancy group.

Incident: An RrgMultipleSecondary incident is generated. The correlation nature of RrgMultipleSecondary is service impact.

Status: The status of the router redundancy group is set to Minor.

Conclusion: RrgMultipleSecondary

Router redundancy group has degraded



Scenario: The router redundancy group has experienced some change. The group is functioning, and there is one primary router and one secondary router, but there is some non-normal condition that could be an issue. For example, there might be several routers not in Standby state.

Root Cause: This scenario could be due to some misconfiguration of the router redundancy group.

Incident: An RrgDegraded incident is generated. The correlation nature of RrgDegraded is service impact.

Status: The status of the router redundancy group is set to Warning.

Conclusion: RrgDegraded

(26) Node component scenarios

Fan failure or malfunctioning

Scenario: A fan sensor detects a failed fan in a chassis.

Incident: A FanOutOfRangeOrMalfunctioning incident is generated.

Status: The status of the fan sensor node component is Critical. The status of Major is propagated to the node.

Conclusion: FanOutOfRangeOrMalfunctioning

Power supply failure or malfunctioning

Scenario: A power supply sensor detects a failed power supply in a chassis.

Incident: A PowerSupplyOutOfRangeOrMalfunctioning incident is generated.

Status: The status of the power supply node component is Critical. The status of Major is propagated to the node.

Conclusion: PowerSupplyOutOfRangeOrMalfunctioning

Temperature exceeded or malfunctioning

Scenario: A temperature sensor detects a high temperature in a chassis.

Incident: A TemperatureOutOfRangeOrMalfunctioning incident is generated.

Status: The status of the temperature sensor node component is Critical. The status of the node does not change.

Conclusion: TemperatureOutOfRangeOrMalfunctioning

Voltage out of range or malfunctioning

Scenario: A voltage sensor detects a voltage problem in a chassis.

Incident: A VoltageOutOfRangeOrMalfunctioning incident is generated.

Status: The status of the voltage sensor node component is Critical. The status of the node does not change.

Conclusion: VoltageOutOfRangeOrMalfunctioning

D.7 Network configuration changes

During the course of a day, an NNMi operator might complete several configuration changes. The following scenarios illustrate some common network configuration changes and show how NNMi responds to these changes.

D. The Causal Engine and NNMi Incidents

(1) 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 completes the following tasks:

- Recalculates the status of the node.
- Closes all registered incidents for the deleted IPv4 addresses and interfaces on the node.

(2) 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 recalculates the status of the connection.

(3) 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 for some vendor-specific link traps. These notifications trigger the NmsApa service to initiate a rediscovery of device information from the node that generated the trap.

Note

For a complete list of the trap incident configurations that NNMi provides, see NNMi Help or select **SNMP Trap Configurations** from **Incidents** in the **Configuration** workspace.

D.8 NNMi management configuration changes

During the course of a day, an NNMi administrator might complete several NNMi configuration changes. The following scenarios illustrate some common NNMi management configuration changes and show how NNMi responds to these changes.

• The NNMi administrator does not manage an IP address or puts it out-of-service

The NmsApa service receives a notification from State Poller after the pingState is set to Not Polled. The NmsApa service sets the status of the IPv4 address to No Status.

• The NNMi administrator manages an IPv4 address or puts it back in service

The NmsApa service receives a notification from State Poller after the pingState is set to the measured value. The NmsApa service calculates the status of the IPv4 address based upon the measured value.

- The NNMi administrator does not manage an interface or puts it out-of-service The NmsApa service receives a notification from State Poller after the operState is set to Not Polled. In response to this notification, the NmsApa service sets the status of the interface to No Status.
- The NNMi administrator manages an interface or puts it back in service

The NmsApa service receives a notification from State Poller after the operState is set to the measured value. The NmsApa service calculates the status of the interface based upon the measured value.

• The NNMi administrator does not manage a node or puts it out-of-service

The NmsApa service receives a notification from State Poller 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. In response to this notification, the NmsApa service sets the status of the node to No Status.

• The NNMi administrator manages a node or puts it back in service

The NmsApa service receives a notification from State Poller 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. In response to this notification, the NmsApa service calculates the status of the node.

D. The Causal Engine and NNMi Incidents

E. List of Ports Used by NNMi

The following table lists the ports NNMi uses on the management server. NNMi listens on these ports. If port conflicts occur, you can change most of these port numbers as shown in the *Change configuration* column.

Important

For application failover to work successfully, use the following configurations:

- Open TCP ports 7800 to 7810.
- The active and standby NNMi management servers must have unrestricted network access to each other.

To run NNMi in a cluster system in an HA configuration, the port number configurations used on the primary and secondary cluster nodes must be the same. When you change a port in the nms-local.properties file, you must configure each node (the port change is not copied by file replication of the HA configuration).

Table E-1: Ports used on the NNMi management server

Port	Туре	Name	Purpose	Change configuration
80	ТСР	nmsas.server.port.we b.http	 Default HTTP port Used for Web UI & Web Services. In GNM configurations, NNMi uses this port to establish communication from the global manager to the regional managers. Once this port is open, it becomes bi-directional. 	<pre>Modify the nms-local.properties file. You can also change this during installation. Windows %NNM_CONF%\nnm\props\nms- local.properties Linux \$NNM_CONF/nnm/props/nms- local.properties</pre>
162	UDP	trapPort	SNMP trap port	Modify using the nnmtrapconfig.ovpl Perl script.
443	ТСР	nmsas.server.port.we b.https	 Default secure HTTPS port (SSL) Used for Web UI & Web Services. 	<pre>Modify the nms-local.properties file. You can also change this during installation. Windows %NNM_CONF%\nnm\props\nms- local.properties Linux \$NNM_CONF/nnm/props/nms- local.properties</pre>
1098	ТСР	nmsas.server.port.na ming.rmi	 Used by NNMi command line tools to communicate with a variety of services used by NNMi. We recommend configuring the system firewall to restrict access to this port to local host only. 	<pre>Modify the nms-local.properties file. Windows %NNM_CONF%\nnm\props\nms- local.properties Linux \$NNM_CONF/nnm/props/nms- local.properties</pre>
1099	ТСР	nmsas.server.port.na ming.port	• Used by NNMi command line tools to communicate with a variety of services used by NNMi.	Modify the nms-local.properties file. • Windows
Port	Туре	Name	Purpose	Change configuration
------	------	---------------------------------------	---	--
1099	ТСР	nmsas.server.port.na ming.port	• We recommend configuring the system firewall to restrict access to this port to local host only.	<pre>%NNM_CONF%\nnm\props\nms- local.properties Linux \$NNM_CONF/nnm/props/nms- local.properties</pre>
3873	ТСР	nmsas.server.port.re moting.ejb3	 Used by NNMi command line tools to communicate with a variety of services used by NNMi. We recommend configuring the system firewall to restrict access to this port to local host only. 	<pre>Modify the nms-local.properties file. Windows %NNM_CONF%\nnm\props\nms- local.properties Linux \$NNM_CONF/nnm/props/nms- local.properties</pre>
4444	ТСР	nmsas.server.port.jm x.jrmp	 Used by NNMi command line tools to communicate with a variety of services used by NNMi. We recommend configuring the system firewall to restrict access to this port to local host only. 	<pre>Modify the nms-local.properties file. Windows %NNM_CONF%\nnm\props\nms- local.properties Linux \$NNM_CONF/nnm/props/nms- local.properties</pre>
4445	ТСР	nmsas.server.port.jm x.rmi	 Used by NNMi command line tools to communicate with a variety of services used by NNMi. We recommend configuring the system firewall to restrict access to this port to the local host only. 	<pre>Modify the nms-local.properties file. Windows %NNM_CONF%\nnm\props\nms- local.properties Linux \$NNM_CONF/nnm/props/nms- local.properties</pre>
4446	ТСР	nmsas.server.port.in voker.unified	 Used by NNMi command line tools to communicate with a variety of services used by NNMi. We recommend configuring the system firewall to restrict access to this port to local host only. 	<pre>Modify the nms-local.properties file. Windows %NNM_CONF%\nnm\props\nms- local.properties Linux \$NNM_CONF/nnm/props/nms- local.properties</pre>
4457	ТСР	nmsas.server.port.hq	 Used for un-encrypted global network management traffic. Messaging travels from the global manager to the regional managers. Once this port is open, it becomes bi-directional. 	<pre>Modify the nms-local.properties file. Windows %NNM_CONF%\nnm\props\nms- local.properties Linux \$NNM_CONF/nnm/props/nms- local.properties</pre>
4459	ТСР	nmsas.server.port.hq .ssl	 Used for encrypted global network management traffic. Messaging travels from the global manager to the regional managers. 	<pre>Modify the nms-local.properties file. Windows %NNM_CONF%\nnm\props\nms- local.properties</pre>

Port	Туре	Name	Purpose	Change configuration
4459	ТСР	nmsas.server.port.hq .ssl	• Once this port is open, it becomes bi-directional.	• Linux \$NNM_CONF/nnm/props/nms- local.properties
4712	ТСР	nmsas.server.port.ts .recovery	Internal transaction service port	<pre>Modify the nms-local.properties file. Windows %NNM_CONF%\nnm\props\nms- local.properties Linux \$NNM_CONF/nnm/props/nms- local.properties</pre>
4713	ТСР	nmsas.server.port.ts .status	Internal transaction service port	<pre>Modify the nms-local.properties file. Windows %NNM_CONF%\nnm\props\nms- local.properties Linux \$NNM_CONF/nnm/props/nms- local.properties</pre>
4714	ТСР	nmsas.server.port.ts .id	Internal transaction service port	<pre>Modify the nms-local.properties file. Windows %NNM_CONF%\nnm\props\nms- local.properties Linux \$NNM_CONF/nnm/props/nms- local.properties</pre>
5432	ТСР	com.hp.ov.nms.postgr es.port	This PostgreSQL port is the port the embedded database listens on for this NNMi management server.	<pre>Modify the nms-local.properties file. Windows %NNM_CONF%\nnm\props\nms- local.properties Linux \$NNM_CONF/nnm/props/nms- local.properties</pre>
5447	ТСР	trapReceiverNettyPor t	Port that stands by for a connection from JBoss. This port is used by TrapReceiver.	Modify with the nnmtrapconfig.ovpl Perl script.
7500	UDP	nnmcluster	Port used by nnmcluster.	Configuration cannot be modified.
7800 to 7810	ТСР		 JGroups ports used for application failover. If application failover is not used, we recommend configuring the system firewall to restrict access to these ports. 	<pre>Modify the nms- cluster.properties file. • Windows %NNM_CONF%\nnm\props\nms- cluster.properties • Linux \$NNM_CONF/nnm/props/nms- cluster.properties</pre>
8886	ТСР	OVsPMD_MGMT	NNMi ovspmd (process manager) management port	1. Run the ovstop command to stop the NNMi service.

Port	Туре	Name	Purpose	Change configuration
8886	ТСР	OVsPMD_MGMT	NNMi ovspmd (process manager) management port	<pre>2. Open the services file: - Windows: %Windir%\system32\drivers \etc\services - Linux: /etc/services 3. Add the following line to the file: ovspmd_mgmt port-number/tcp 4. Run the ovstart command to start the NNMi service.</pre>
8887	ТСР	OVsPMD_REQ	NNMi ovsmpd (process manager) request port	 Run the ovstop command to stop the NNMi service. Open the services file: Windows: Windir%\system32\drivers \etc\services Linux:
8989	ТСР	<pre>com.hp.ov.nms.events .action.server.port</pre>	Enables an action server port so that it can be configured.	<pre>Modify the nnmaction.properties file: Windows %NnmDataDir%shared\nnm\conf \props\nnmaction.properties Linux \$NnmDataDir/shared/nnm/ conf/props/ nnmaction.properties</pre>

Legend:

--: No name

Table E-2 lists some of the ports NNMi uses to communicate with other systems. If a firewall separates NNMi from these systems, open many of these ports in the firewall. The actual set of ports to open depends on the system to be linked to NNMi and how that system is configured.

Table E-2: Firewall pass-through direction

Purpose	Port No. (Port/Type)	Firewall pass-through direction
NNMi console	80/tcp	 NNMi ← Web browser NNMi (global manager) → NNMi (regional manager)
SNMP request	161/udp	NNMi → Monitored node
SNMP response	ANY/udp	NNMi ← Monitored node ^{#1}
SNMP trap/SNMP inform request	162/udp	NNMi 🗲 Monitored node
Response to SNMP inform request	ANY/udp	NNMi \rightarrow Monitored node ^{#2}

Purpose	Port No. (Port/Type)	Firewall pass-through direction
SNMP trap transfer	162/udp	NNMi \rightarrow SNMP manager NNMi \rightarrow Northbound application
LDAP	389/tcp	NNMi \rightarrow LDAP server
NNMi console using SSL connection	443/tcp	 NNMi ← Web browser NNMi (global manager) → NNMi (regional manager)
LDAP using SSL connection	636/tcp	NNMi \rightarrow LDAP server
Messaging bisocket connector	4457/tcp	NNMi (global manager) 🔿 NNMi (regional manager)
Messaging bisocket connector using SSL connection	4459/tcp	NNMi (global manager) → NNMi (regional manager)
Application failover	7800 to 7810/tcp	NNMi (active) $\leftarrow \rightarrow$ NNMi (standby)

Legend:

 $\leftarrow \rightarrow$:

For TCP, the arrow indicates the direction in which a connection is made.

For UDP, the arrow indicates the direction in which a packet is sent.

#1: An SNMP response is made from the SNMP request receiving port to the SNMP request sending port.

#2: A response to an SNMP inform request is made from the SNMP inform request receiving port to the SNMP inform request sending port.

Notes:

1. You must configure the firewall to pass ICMP between NNMi and the monitored node.

2. The port numbers are set to the default settings.

3. For details about configuring application failover, see 18. Configuring NNMi for Application Failover.

If you configure NNMi to use ICMP fault polling or ping sweep for node discovery, configure the firewall to pass ICMP packets through the firewall.

If you plan to use the global network management feature, Table E-3 shows the ports that need to be accessible to a regional NNMi management server from a global NNMi management server. The global network management feature requires these ports to be open for TCP access from the global NNMi management server to the regional NNMi management server. The regional NNMi management server will not open sockets back to the global NNMi management server.

Table E-3: F	Required	accessible	sockets	for global	network n	nanagement

Security	Parameter	TCP port
Non-SSL	nmsas.server.port.web.http	80
	nmsas.server.port.hq	4457
SSL	nmsas.server.port.web.https	443
	nmsas.server.port.hq.ssl	4459

F.1 Changes in version 12-50

(1) Changes in manual 3021-3-E02-20(E)

- A description about kernel.shmall was added to the preinstallation checklist for the NNMi management server.
- A description about the specification of HTTPS port numbers during the installation of NNMi was added.
- LANG was added as an environment variable to be specified during the installation of NNMi.
- A description about supported certificate formats was added.
- The procedure for replacing a certificate that was created during the installation of NNMi was changed.
- A description about cases where a root CA certificate needs to be imported for multiple LDAP directory servers was added.
- Microsoft Edge was added as a browser to be configured for a telnet or SSH client.
- The description about Windows operating systems that run Internet Explorer or Firefox was changed.
- A description about server elements was changed.
- A description about the default port numbers set by the NNMi installation script was added.
- The expression for enabling the function that auto-trims the oldest incidents was changed.
- The expression for enabling the function that auto-trims the oldest SNMP trap incidents was changed.
- A description about the rotation of archive files was added.
- A description in *Upgrading from NNMi Version 9, 10, or 11*, was changed to explain how to upgrade to version 12-50.
- In the list of ports used by the NNMi management server, a description about changing the settings of port 443 was added.

F.2 Changes in version 12-10

(1) Changes in manual 3021-3-E02-10(E)

- The description of the NNMi management server pre-installation checklist was changed.
- A description of the Compatibility View Setting when Internet Explorer is used was added.
- The following items were added as the packages required by NNMi:
 - fontconfig package
 - liberation-sans-fonts package
 - libnsl package

F. Version Changes

JP1/Network Node Manager i Setup Guide

- A description of checking the completion of NNMi installation in a Windows system was added.
- The following directory was deleted from the files and directories that are excluded from the scan target:
 - /etc/opt/OV
- Descriptions of distribution where systemd is used to manage services were added.
- Descriptions of the automatic trimming setting for incidents were added or changed.
- Descriptions of the directories and files to be deleted after NNMi is removed were changed.
- In the procedure for generating a self-signed certificate, the command that generates a private key from the system was changed.
- In the procedure for generating a CA-signed certificate, the command that imports the certificate into a keystore file was changed. In accordance with this change, the note about the command was changed.
- A description of the owner CN of the Truststore certificate was added.
- A description in *Upgrading from NNMi Version 9, 10, or 11*, was changed to explain how to upgrade to version 12-10.
- A description of linking with JP1/IM2 Intelligent Integrated Management Base was added.
- Descriptions of the environment variables used in this manual were added or deleted.
- The description of the NodeOrConnectionDown incident was changed.
- The following OSs are now supported:
 - CentOS 8.1 or later
 - Linux 8.1 or later
 - Oracle Linux 8.1 or later
 - Windows Server 2019

F.3 Changes in version 12-00

(1) Changes in manual 3021-3-E02(E)

- Windows Server 2008 R2 was removed from the applicable OSs.
- The description of the NNMi management server preinstallation checklist was changed.
- The note on the settings of Internet Explorer when enabling a Web browser for the NNMi console was changed.
- The following item was added under Installing NNMi (Windows):
 - Check the installation results.
- The following items was added as tasks to be performed after installing NNMi:
 - Exclude NNMi files and directories from the scan target
- The description of how to set the language environment after installing NNMi on a Linux server was changed.
- A description about how to switch from trial licenses of NNMi and NNMi Advanced to upper-level trial licenses was added.
- In *Removing NNMi (Windows)*, the following directories and files were added to the temporary directories and temporary files that must be deleted:

- %TEMP%\MicroFocusOvInstaller\
- %TEMP%\NNM_X.X.X_MicroFocusOvInstaller.txt
- %TEMP%\ovinstallparams.ini
- %TEMP%\nnmilog\
- drive:\ProgramData\apregid.com.hpe
- In *Removing NNMi (Linux)*, the following directories and files were added to the temporary directories and temporary files that must be deleted:
 - /var/tmp/jplnnmi
 - /tmp/MicroFocusOvInstaller
 - /tmp/NNM_X.X.X_MicroFocusOvInstaller.txt
 - /usr/share/apregid.com.hpe
- Descriptions about detailed information of NNMi Help were deleted.
- The title *Configuring Communication for Virtual Environments* was changed to *Discover and monitor VMware hypervisor-based virtual networks*.
- The notes and example of the configuration required to discover and monitor virtual networks on hypervisors were deleted. Also, the description about the procedure for replacing the VMware default certificate was changed to a reference to the VMware documentation.
- The procedure for enabling the TLSv1 cryptographic protocol on an NNMi management server was deleted from the descriptions of *Configuring NNMi to Communicate with Hypervisors Using HTTPS*.
- In Configuring communications, the following descriptions were added:
 - Discover and monitor Cisco ACI networks
 - Multihomed NNMi Management Server
- The description of details about the concept of incidents was changed to a reference to the NNMi Help.
- The default value of the com.hp.ov.nms.trapd.recvSocketBufSize trap server property was changed.
- The following descriptions were added or changed in the procedure for generating a CA-signed certificate:
 - The note about the command for checking the contents of a CSR file was added.
 - The content of the trust store output format was changed.
- The storepass option was added to the example of accessing the trust store by executing the nnmkeytool.ovpl command.
- Descriptions about the ldap.properties file were changed or deleted, because the file has been removed.
- The following item was deleted from the descriptions of the functions that are not supported by the NNMi IPv6 management function:
 - Discovery of IPv6 subnet connections
- A condition regarding IPv4 addresses was added to the prerequisite conditions for setting up application failover.
- A note was added to the procedure for setting up application failover when either node has multiple IPv4 addresses or NICs.
- A note was added for the situation when the mount point of a shared disk is changed after HA configuration.
- The following description was deleted from Maintaining NNMi:
 - Configuring NNMi to enforce strict SNMPv3 inform processing

- The following description was deleted from NNMi Security:
 - Configuring NNMi to stop reporting the ovjboss version number
- A description in *Upgrading from NNMi Version 9, 10, or 11*, was changed to explain how to upgrade to version 12-00.

F.4 Changes in version 11-50

(1) Changes in manual 3021-3-A72-20(E)

- Descriptions of how the installation script behaves and how to handle the behavior were added as items to check before installing NNMi on an NNMi management server.
- The unzip command was added as a command required to install NNMi on a Linux server.
- In Configuring communications, the following descriptions was changed:
 - *Configuring an SNMP proxy*
- A procedure for using the TLSv1 cryptographic protocol in an environment where a new installation of NNMi 11-50 was performed was added.
- In *Working with Certificates for NNMi*, descriptions were changed to include the use of certificate repositories in the PKCS #12 format and the JKS format. The following descriptions were deleted accordingly:
 - Configuring the application failover feature to use CA certificates
 - Configuring the global network management feature to use a Certificate Authority
- Certificate repositories in the PKCS #12 format are now supported, and descriptions were added or changed accordingly.
- In *Working with Certificates for NNMi*, descriptions were changed to include the use of certificate repositories in the JKS format. In addition, the commands to be run were changed in the following topics:
 - *Generating a Self-Signed Certificate*
 - Generating a CA-Signed Certificate
 - Configuring application failover to use self-signed certificates
 - Configuring an HA cluster to use a new certificate
 - Configuring an SSL connection to the directory service
- The file nms-auth-config.xml was added as an LDAP configuration file. Accordingly, the related descriptions were added, and the description of the existing LDAP configuration file ldap.properties was changed.
- The following descriptions were deleted from *Task 3: Configure user access from the directory service* and *Task 5: (Configuring for the external mode only) Configure group retrieval from the directory service*:
 - Simple approach for Microsoft Active Directory
 - Simple approach for other directory services
- A note about a message that can be ignored in mixed mode when settings are specified for the LDAP configuration file was added.
- A command to be run when the value of the defaultRole parameter is changed was added.
- The following descriptions were deleted from User identification.
 - Configuring NNMi user access from the directory service (detailed approach)
 - Determining how the directory service identifies a user (LDAP browser approach)

F. Version Changes

- Determining how the directory service identifies a user (Web browser approach)
- A procedure for switching the LDAP configuration file from the file ldap.properties to the file nms-auth-config.xml was added.
- In the subsection *Initial preparation* (in *Global Network Management*,), the description for configuring certificates was changed. In addition, the following description was added:
 - NNMi management servers upgraded to the version 11-50
- A description of HTTP access to NNMi was added.
- A procedure for performing a restoration in an NNMi failover environment on a different set of servers was added. In addition, notes were added regarding the backups that are necessary when performing such a restoration.
- In Configuring NNMi in a High Availability Cluster, the following descriptions were changed:
 - Changing virtual IP addresses
 - Renaming physical hosts
 - Unconfiguring NNMi from an HA cluster
 - Unconfiguring NNMi on the passive cluster node
 - Unconfiguring NNMi on the active cluster node
- In Maintaining NNMi, the following descriptions were changed:
 - Configuring communication settings
 - Configuring NNMi to require encryption for remote access
- The description of specifying a file path on the Windows NNMi management server was changed.
- In Administering SNMP traps, the following description was added:
 - Block SNMPv1 or SNMPv2c Traps
- In the subsections that describe *enabling the auto-trim oldest SNMP trap incidents feature* (in *Maintaining NNMi*), the values used in the description were changed.
- In the subsection *Configuring physical sensor status*, the text to be added to the property file was changed in the following topics:
 - Propagating physical sensor status to a physical component
 - Configuring physical sensor status to not propagate to the physical component
 - Overriding physical sensor status values
- In NNMi Security, the following description was changed:
 - Configure TLS Protocols
- A description in *Upgrading from NNMi Version 9, 10, or 11*, was changed to explain how to upgrade to version 11-50. In addition, procedures to be performed after upgrading to version 11-50 were added to the following topics:
 - Global network management upgrade steps
 - Upgrading to NNMi 11-50 configured for application failover
- Chapter 29. RESTfulAPI was added to Part 8: Integration with NNMi.
- Descriptions were added for the environment variables %jdkdir% and \$jdkdir.

F.5 Changes in version 11-10

(1) Changes in manual 3021-3-A72-10(E)

- Windows Server 2016 is now supported.
- The following file was deleted from the library files required when installing NNMi on a Linux server:
 - /usr/lib/libstdc++.so.6
- The description of how to set the language environment after installing NNMi on a Linux server was changed.
- Notes on tracking license information were added.
- The following description was added to *Initial startup problems*:
 - You cannot start the NNMi console when accessing a Windows NNMi management server
- A description was added that explains that, if SNMP Agents and Web Agents are configured, NNMi can use additional protocols.
- In SNMP access control, the description of SNMPv3 privacy protocols was changed.
- A description was added that explains that, to discover hypervisors, NNMi requires the node name rather than the management address.
- A description of configuring communication for virtual environments was added.
- A description for VMware communication was added to Checking the communication settings.
- A description of when the number of nodes discovered by NNMi reaches or exceeds the licensed capacity limit was added to *Concepts of discovery*.
- A description that explains conditions where virtual machine nodes are not deleted was added to *Deleting unresponsive objects*.
- The following descriptions were added or changed in *Planning state polling*:
 - What Can NNMi Monitor?
 - Stop Monitoring
 - Interfaces to Unmonitored Nodes
 - Extending Monitoring
- In the description of node groups, virtual machines were added to node groups provided by default.
- Information of "Web Polling" was added to the information to be collected by the State Poller service.
- "State updater exceptions" was added to the table "State Poller health information".
- The description of "What to forward", which is explained in the table "Supported ways to forward traps and NNMi incidents", was changed.
- In NNMi Console, the following descriptions were added or changed:
 - Configuring Gauges in the Analysis Pane
 - Configuring Map Label Scale Size and Borders
 - Configuring Auto-Collapse Thresholds for Loom and Wheel Diagrams
- In Working with Certificates for NNMi, the following descriptions were added or changed:
 - About NNMi Certificates
 - Replacing an Existing Certificate with a new Self-Signed or CA-Signed Certificate
- In Maintaining NNMi, the following descriptions were added or changed:
 - Modifying MIB Browser Parameters
 - Configuring NNMi to allow level 2 operators to delete nodes and incidents

F. Version Changes

- Configuring NNMi to allow level 2 operators to edit maps
- The following description was added to *NNMi Security*:
 - Configure TLS Protocols
- The following description was added to Upgrading from NNMi Version 9, 10, or 11:
 - Upgrading NNMi management servers from version 11-00
- The following descriptions were added to Migrating NNMi from an HP-UX or Solaris Operating System:
 - Changing NNMi from HP-UX or Solaris to Linux on Application Failover environment
 - Changing NNMi from HP-UX or Solaris to Linux on Application Failover environment on the global manager and regional managers
 - Changing NNMi from HP-UX or Solaris to Linux in a High Availability Cluster
- The following MIBs were deleted from those that are read during a new installation:
 - AX-BFD-MIB
 - AX-BOOTMANAGEMENT-MIB
 - AX-DEVICE-MIB
 - AX-FDB-MIB
 - AX-FLOW-MIB
 - AX-LOGIN-MIB
 - AX-MANAGEMENT-MIB
 - AX-NOTIFICATION
 - AX-OSPF-MIB
 - AX-OSPFV3-MIB
 - AX-QUEUE-MIB
 - AX-SMC-MIB
 - AX-SMCSERVICE-MIB
 - AX-SMI-MIB
 - AX-STATS-MIB
 - AX-SYSTEM-MIB
 - AX-TRACK-MIB
 - AX-VLAN-MIB
 - AX-VRF-MIB
 - AX1230S
 - AX1240S
 - AX2000R
 - AX2430S
 - AX2530S
 - AX3630S
 - AX4630S
 - AX5400S-TRAP

F. Version Changes

- AX6300S
- AX7700R-TRAP
- AX7800R
- AX7800R-TRAP
- AX7800S
- AX7800S-TRAP
- AXS-6700S-TRAP
- AXS-AX1240S-TRAP
- AXS-AX1250S-TRAP
- AXS-AX2230S-TRAP
- AXS-AX3630S-TRAP
- AXS-AX3640S-TRAP
- AXS-AX3650S-TRAP
- AXS-AX3830S-TRAP
- AXS-AX4630S-TRAP
- AXS-AX6300S-TRAP
- AXS-AX6600S-TRAP
- Apresia-Series
- Apresia-SeriesLightFMGM
- BFD-TC-STD-MIB
- COMETAGT-AIX
- COMETAGT-LINUX
- COMETAGT-SOLARIS
- COMETAGT-TRU64
- RFC1253-MIB
- cmSmsAgt
- cometAgt
- cometAgtEx
- windowsNTAgt
- The following environment variables were deleted from the extended list of NNMi environment variables:
 - %NNM_SUPPORT% (Windows)
 - \$NNM_SUPPORT (Linux)

F.6 Changes in version 11-00

(1) Changes in manual 3021-3-A72(E)

• The following OSs are now supported:

- Windows Server 2008 R2 (x64) SP2
- CentOS 6.1 or later
- CentOS 7.1 or later
- Linux 6.1 (x64) or later
- Linux 7.1 or later
- Oracle Linux 6.1 or later
- Oracle Linux 7.1 or later
- SUSE Linux 12
- The following OSs are no longer supported:
 - HP-UX (IPF)
 - Solaris
- Chapter 1. Preinstallation Checklists and Chapter 2. Installing and Uninstalling NNMi were added to Part 1: Preparation.
- Chapter 3. Getting Started with NNMi was added to Part 2: Introduction.
- Notes about using SNMPv3 in NNMi to communicate with devices were added.
- Appendix A. When NNMi Manpages Cannot Be Displayed (Linux) and Appendix B. List of MIBs Read During a New Installation were added.
- A description of discovery using link aggregation was added.
- A description of how to poll devices by NNMi when SNMP traps are received was added.
- The description of how to implement NAT in NNMi was entirely rewritten.
- A description of how to configure monitoring for static NAT and dynamic NAT was added.
- The procedure for deploying NNMi in a network address translation (NAT) environment was added.
- A description of the SNMP agent status calculations determined by the combined ICMP and SNMP responses when management address polling is enabled was added.
- A description of replicating custom attributes from a regional manager to the global manager was added.
- The procedure for reactivating the IPv6 feature was added.
- The method for automating the configuration of node groups by using a command line tool was added.
- The method for configuring node group maps by using a command line tool was added.
- The method for configuring communication settings by using a command line tool was added.
- A description of how to override settings in the server.properties file was added.
- The procedure for configuring NNMi to authenticate SNMPv3 traps for nodes that either are managed using SNMPv2 or SNMPv1 or are not discovered, and the procedure for configuring timeframes within which the Causal Engine accepts traps were added.
- The procedure for determining the original trap addresses from traps sent by a proxy SNMP gateway was added.
- A description of the order of SNMPv1 and SNMPv2c trap addresses was added.
- A description of the standalone NmsTrapReceiver process to help minimize the loss of SNMP traps during a failover was added.
- A description of how to schedule outages for an arbitrary set of nodes using NNMi was added.
- The procedure for configuring physical sensor status and node sensor status was added for monitoring status.

F. Version Changes

- The procedure for configuring NNMi to enable or disable SSLv3 ciphers was added.
- A description of NNMi data encryption was added.

F.7 Changes from version 10-10 to version 10-50

(1) Changes in manual 3021-3-343-20(E)

- The following OSs are now supported:
 - Microsoft^(R) Windows Server^(R) 2012 R2 Datacenter
 - Microsoft^(R) Windows Server^(R) 2012 R2 Standard
- Because it is not possible in Windows Server 2008 and later to log in to a console session from a remote desktop, the associated descriptions were removed.
- The description of the NNMi management server preinstallation checklist was changed.
- The following item was added under Installing NNMi (Windows):
 - Confirmation of whether to perform the preinstallation checks and continue the installation
- The following item was added under Installing NNMi (UNIX):
 - Confirmation of whether to perform the preinstallation checks and continue the installation
- The following items were added as tasks to be performed after installing NNMi:
 - Set the language environment (UNIX only)
 - Check the maximum Java heap size
- The steps listed under *Specifying disk drive security settings (Windows)* were changed.
- The following MIBs were added to the list of MIBs that are read during a new installation:
 - AX-BOOTMANAGEMENT-MIB
 - AX-DEVICE-MIB
 - AX-FLOW-MIB
 - AX-LOGIN-MIB
 - AX-NOTIFICATION
 - AX-OSPF-MIB
 - AX-OSPFV3-MIB
 - AX-QUEUE-MIB
 - AX-SMI-MIB
 - AX-STATS-MIB
 - AX-SYSTEM-MIB
 - AX-VRF-MIB
- A limitation related to use of interface groups in the Discovery Excluded Interfaces configuration was added.
- The description of device support when NETCONF is used in the communication configuration was added.
- The description of how to use the excluded IP addresses feature to exclude certain objects from discovery was changed.

F. Version Changes

- A description of the NNMi Northbound interface was added.
- Changes were made to the parameters of the command used to generate a private key from the system when a Certificate Authority certificate is generated.
- The description of integrating NNMi with a Directory Service when LDAP is used was changed.
- The description of the maps and path views in the section on the effects of limiting object access was changed.
- The description of how to configure the application failover feature was changed.
- A command for restoring in the same cluster a deleted standby server in the NNMi database in an application failover configuration was added.
- A description was added regarding application failover control during a restart after a communication failure.
- The Symantec Cluster Server (SCS) HA cluster software was added.
- Notes about HA configurations were added.
- An example of a configuration of WSFC resources was added.
- A description of enabling actions for secondary root cause management events was added.
- The items to be created or modified with a new author specified were changed.
- A description of merging SSL certificates when the NNMi configuration and database are moved between systems was added.
- The description regarding changing the host name or domain name of an NNMi management server was changed.
- Descriptions regarding the following NNMi security items were added:
 - Providing a password for embedded database tools
 - Configuring NNMi to stop reporting the ovjboss version number

F.8 Changes from version 10-00 to 10-10

(1) Changes in manual 3021-3-343-10(E)

- Descriptions for Windows Server 2012 were added.
- Notes about using network address translation (NAT) for discovering nodes were added.
- Changes were made to the Quick Start Configuration Wizard window and its description.
- The description of uninstalling NNMi was changed.
- Changes were made to the NNMi Help window and the **Communication Configuration** form.
- Changes and additions were made to the description of disk drive security settings for Windows.
- The descriptions of the settings and procedures for enabling Web browsers were changed.
- Additions and changes were made to the description of installing required libraries in Linux.
- Information about the following installation problem was added:
 - A message is displayed during installation indicating that the preinstallation procedure (phase II) has failed and the /tmp/nnm_preinstall_phaseII.log file needs to be checked for the details.
- Additions and changes were made to the list of MIBs read during a new installation.
- The description of how to use the Author attribute was changed.
- A description concerning menus and menu items was changed.

F. Version Changes

- Notes were added about using the nnmconfigimport.ovpl command to import a large number of settings.
- Information was added about the following aspects of SNMP communication:
 - Changing the encryption method when using SNMPv3 communication
 - Enabling or disabling a specific device's SNMP communication
 - SNMP communication procedure using an SNMP proxy agent
- Information was added about discovery using tenants in a network that contains overlapping address domains.
- Methods for specifying excluded IP addresses and for specifying excluded interface groups were added to the settings for not discovering objects.
- A description was added of how to specify a range of interfaces to be discovered by defining a filter.
- For handling seed discovery errors, a description was added of how to include the corresponding IP addresses in the ipnolookup.conf file.
- The description of deleting non-responding objects was changed.
- In NNMi state polling, the following items were changed:
 - Proxy servers were changed to switches.
 - The description of how to set the interface groups and node groups to be monitored was changed.
- Regarding NNMi incidents, descriptions of the following were added or changed:
 - Concepts of incidents
 - Trap and incident forwarding
 - Received SNMP traps
 - CIAs added to closed management event incidents
 - Planning how NNMi is to respond to incidents
 - How to set trap logs
 - How to set incident logs
 - How to set trap server properties
 - Batch loading of incident configurations
 - Evaluating incidents
 - Tuning incidents
- Regarding the NNMi console, descriptions of the following were added or changed:
 - Creating node groups
 - Configuring node group maps
 - Deleting node groups
 - Disabling the Analysis pane
 - Customizing device icons
 - Overriding the refresh rate of table views
- In the method for using certificates in NNMi, the following procedure was changed:
 - Generating a Certificate Authority certificate
- The description of configuring an SSL connection to the directory service was changed.
- In integrating NNMi with a directory service through LDAP, the descriptions of the following were changed:

F. Version Changes

- NNMi user access information and configuration options
- Configuring NNMi to access a directory service
- Changing the directory service access configuration to support the NNMi security model
- Information owned by the directory service administrator
- Troubleshooting the directory service integration
- It was made clear that in the NNMi global operator user group (globalops), access permissions are granted only to all topology objects.
- A chapter on how to configure a NAT environment was added.
- The description of NNMi security and multi-tenancy configuration was changed.
- In global network management, notes about NAT, PAT, and NAPT were added.
- For the firewall configuration in the initial preparations, the parameters for the required accessible sockets were changed.
- For purposes of global network management, the NNMi window and its description were changed.
- In troubleshooting tips for global network management, the following description was changed:
 - Synchronizing regional manager discovery from a global manager
- The description of upgrading NNMi in a global network management environment was changed.
- A description about global network management and the address translation protocol was added.
- Regarding the NNMi IPv6 management feature, the following descriptions were added or changed:
 - Overview of the NNMi IPv6 management feature
 - Prerequisites for using the NNMi IPv6 management feature
 - Activating the IPv6 management feature capabilities
 - IPv6 inventory following deactivation of the IPv6 management feature
- Prerequisites for setting up application failover were added.
- The following application failover setup methods were added or changed:
 - Configuring NNMi for application failover
 - Configuring application failover by using the cluster setup wizard
 - Configuring application failover communication
 - Application failover behavior
 - Application failover scenarios
 - Disabling application failover
 - Upgrading NNMi (including applying a patch)
 - Changing the NNMi database password
- The following descriptions concerning the HA configuration were added or changed:
 - Verifying the prerequisites for configuring NNMi for HA
 - NNMi HA configuration information
 - Configuring NNMi on the primary cluster node
 - Configuring NNMi on the secondary cluster node
 - Unconfiguring NNMi on the passive node

F. Version Changes

- Unconfiguring NNMi on the active node
- The description of NNMi data backup was changed.
- The following descriptions about maintaining NNMi were added or changed:
 - Managing folder access permissions
 - Changing the action server queue size
 - Incident actions log
 - Configuring the character set encoding
 - Configuring to allow level 2 operators to delete nodes
 - Configuring to allow level 2 operators to edit maps
 - Configuring to allow level 2 operators to perform state polling and configuration polling
 - Configuring NNMi to authenticate SNMPv3 traps for unmonitored nodes
 - Configuring NNMi to identify the original trap addresses from traps sent by a proxy SNMP gateway
 - · Configuring NNMi to require encryption during remote access
 - Configuring NNMi for strict SNMPv3 inform processing
 - Configuring NNMi to retain the previously supported varbind order
 - Changing the database port
- A change was made to the description of updating the HTTPS configuration with a new certificate when changing the host name and/or domain name of an NNMi management server.
- The following descriptions about upgrading from NNMi version 9 or 10-00 were added:
 - Upgrading a NNMi management server from version 10-00
 - Upgrading a NNMi management server from version 9
 - Upgrading global managers and regional managers from NNMi 10-00 or earlier
 - Upgrading the application failover configuration to NNMi 10-10
- The following descriptions about upgrading from NNMi version 8 or earlier were changed:
 - Configuring SNMP
 - Customizing device profiles
 - Configuring a discovery schedule
 - Setting up auto-discovery rules
 - Specifying a polling interval
 - Selecting a polling protocol
 - Displaying traps from devices
- The default locations of environment variables were changed.
- The list of ports used by NNMi was changed.

This appendix provides reference information, including various conventions, for this manual.

G.1 Related publications

This manual is part of a related set of manuals. The manuals in the set are listed below (with the manual numbers):

- JP1 Version 12 Network Management: Getting Started (3021-3-E01(E))
- JP1 Version 12 JP1/Network Node Manager i Developer's Toolkit Guide (3021-3-E03(E))

G.2 Conventions: Abbreviations for product names

This manual uses abbreviations for Hitachi product names and for the names of products of other companies. The following table shows the abbreviated product names as used in this manual, together with each product's full name.

Abbreviation		Full name or meaning	
Firefox		Mozilla Firefox(R)	
JP1/IM2		JP1/Integrated Management 2	
Linux	CentOS 6.1 (x64)	CentOS 6.1 (x64)	
	CentOS 7.1	CentOS 7.1	
	CentOS 8.1	CentOS 8.1	
	Linux 6.1 (x64)	Red Hat Enterprise Linux(R) Server 6.1 (64-bit x86_64)	
	RHEL 6		
	Linux 7.1	Red Hat Enterprise Linux(R) Server 7.1 (64-bit x86_64)	
	Linux 8.1	Red Hat Enterprise Linux(R) Server 8.1 (64-bit x86_64)	
	RHEL 8.1		
	Oracle Linux 6.1	Oracle Linux(R) Operating System 6.1 (x64)	
	Oracle Linux 7.1	Oracle Linux(R) Operating System 7.1	
	Oracle Linux 8.1	Oracle Linux(R) Operating System 8.1	
	SUSE Linux 12	SUSE Linux(R) Enterprise Server 12	
NNMi	NNMi	JP1/Network Node Manager i	
	NNMi Advanced	JP1/Network Node Manager i Advanced	

G.3 Conventions: Acronyms

This manual also uses the following acronyms:

G. Reference Material for This Manual

Acronym	Full name or meaning
ACL	Access Control List
АРА	Active Problem Analyzer
ARP	Address Resolution Protocol
BIND	Berkeley Internet Name Domain
СА	Certification Authority
CIA	Custom Incident Attribute
DNS	Domain Name System
FQDN	Fully Qualified Domain Name
HSRP	Hot Standby Routing Protocol
НТТР	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Security
ICMP	Internet Control Message Protocol
IPF	Itanium(R) Processor Family
ISP	Internet Services Provider
IT	Information Technology
MD5	Message Digest 5
MIB	Management Information Base
NFS	Network File System
NOC	Network Operations Center
REST	REpresentational State Transfer
SCS	Symantec Cluster Server [#]
SHA	Secure Hash Algorithm
SNMP	Simple Network Management Protocol
TCP/IP	Transmission Control Protocol/Internet Protocol
TLS	Transport Layer Security
UUID	Universally Unique IDentifier
VCS	Veritas Cluster Server [#]
VLAN	Virtual LAN
VRRP	Virtual Router Redundancy Protocol
WAN	Wide Area Network

#

JP1 10-50 or later supports both Veritas Cluster Server and Symantec Cluster Server. JP1 10-10 or earlier supports Veritas Cluster Server only.

G.4 Conventions: KB, MB, GB, and TB

This manual uses the following conventions:

- 1 KB (kilobyte) is 1,024 bytes.
- 1 MB (megabyte) is 1,024² bytes.
- 1 GB (gigabyte) is 1,024³ bytes.
- 1 TB (terabyte) is 1,024⁴ bytes.

G. Reference Material for This Manual

H. Glossary

account

See user account.

active cluster node

The server currently running the NNMi processes in an application failover or high availability configuration.

address hint

See discovery hint.

application failover

An optional NNMi capability that transfers control of NNMi processes to a standby server when the currently active server fails. This feature, which must be configured by the user, utilizes JBoss clustering support.

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.

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, such as status polling, SNMP traps, and changes discovered as a result of certain incidents. Causal Engine uses RCA to examine the status of managed objects, clarifies conclusions related to these objects, and generates root cause incidents.

causality

Denotes the relationship between one event (the cause) and another event (the effect), which is the direct consequence (the 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 by high availability technology or by using JBoss clustering capabilities, that works together to ensure functional and data continuity if components overload or fail. The computers in a cluster are commonly connected to each other through high-speed LANs. Clusters are usually deployed to improve availability, performance, or both.

cluster member or node

In an NNMi context, a system within a high availability or JBoss cluster that has been or will be configured to support NNMi high availability or application failover.

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.

H. Glossary

controller

In NNMi application failover, a JGroups term for the cluster member that has the master cluster state. The controller is always the oldest member of the cluster.

discovery hint

An IP address found by NNMi using 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 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 or system object IDs (object identifiers), or both, used to limit the rule-based discovery process. Discovery rules are configured in the **Discovery Configuration** portion of the NNMi console under **Auto-Discovery Rules**. See also *rule-based discovery*.

discovery seed

See seed.

episode

A term used in NNMi root cause analysis to refer to a specific connection duration, triggered by a primary failure, during which secondary failures are suppressed or are correlated under the primary failure.

fault polling

A key NNMi monitoring activity, in which NNMi issues ICMP pings, SNMP read-only queries of status MIBs, or both, for its managed interfaces, IP addresses, and SNMP agents 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 Settings** in the **Configuration** workspace of the NNMi console. Fault polling is a subset of state polling.

global manager

The NNMi management server in a global network management deployment that consolidates data from distributed NNMi regional manager servers. The global manager provides a unified view of topology and incidents across the entire environment. A global manager must have an NNMi Advanced license.

global network management

A distributed deployment of NNMi with one or more global managers consolidating data from one or more geographically distributed regional managers.

HA

See high availability (HA).

HA resource group

In modern high availability environments, such as Veritas Cluster Server, Symantec Cluster Server, and Microsoft Cluster Service, applications are represented as compounds of resources, such as the application itself, its shared file systems, and a virtual IP address. The resources consist of an *HA resource group*, which represents an application running in a cluster environment.

high availability (HA)

Used in this manual 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 when a component fails. NNMi can be configured to support one of several commercially available HA solutions. See also *application failover*.

ICMP

See Internet Control Message Protocol (ICMP).

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 enable 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 events.

interface

A logical connection terminal for utilizing various specifications and protocols used in networks.

interface group

One of NNMi's primary filtering techniques, where interfaces are grouped together to apply settings to a group or to 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 (ICMP)

One of the core protocols of the Internet protocol suite (TCP/IP). ICMP ping is used by NNMi together with SNMP queries for status polling.

L2

See *Layer 2 (L2)*.

L3

See *Layer 3 (L3)*.

Layer 2 (L2)

Refers to the data link layer of the Open Systems Interconnection (OSI) multilayered communication model. The data link layer moves data across physical links in the network. NNMi Layer 2 views provide information about the physical connectivity of devices.

Layer 3 (L3)

Refers to the network layer of the Open Systems Interconnection (OSI) multilayered communication model. The network layer is concerned with knowing the addresses 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.

Management Information Base (MIB)

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 referred to as *MIB objects*, *objects*, or *MIBs*).

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 (MIB).

NNM events

An NNMi term for events forwarded from previous generation NNM management stations to NNMi. NNMi provides incident views for browsing the incidents that NNMi generates from these forwarded events.

NNMi

Acronym referring to either Job Management Partner 1/Consolidated Management 2/Network Node Manager i or Job Management Partner 1/Consolidated Management 2/Network Node Manager i Advanced.

NNMi is a software product designed to aid network administration and to consolidate network management activities, including ongoing discovery of network nodes, monitoring for events, and network fault management. NNMi is accessed primarily from the NNMi console.

NNMi console

The NNMi user interface. Operators and administrators use the NNMi console for network management tasks.

NNMi management server

A computer system on which the NNMi software is installed and the NNMi processes and services are run.

NNMi Northbound interface

The NNMi functionality that forwards NNMi incidents as SNMPv2c traps to a Northbound application.

node

In the network context, a computer system or device (for example, printer, router, 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 to apply settings to the group or to 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*.

Northbound application

Any application that can receive and process SNMPv2c traps.

Northbound destination

An NNMi Northbound interface configuration that defines a connection to the trap-receiving component of a Northbound application and specifies the types of traps that NNMi sends to that Northbound application.

object identifier (OID)

In SNMP, a numerical sequence that identifies a management information base 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. An OID is the numerical equivalent of an MIB object name; for example, the MIB object name iso.org.dod.internet.mgmt.mib-2.bgp.bgpTraps.bgpEstablished is equivalent to the OID 1.3.6.1.2.1.15.0.1.

OID

See object identifier (OID).

H. Glossary

ovstart command

A command that starts the NNMi managed processes. Invoked at a command prompt. See the ovstart Reference Page.

ovstatus command

A command that reports the current status of the NNMi managed processes. Can be invoked from the NNMi console (by clicking **Tools**, and then **NNMi Status**) or at a command prompt. See the *ovstatus Reference Page*.

ovstop command

A command that stops the NNMi managed processes. Invoked at a command prompt. See the ovstop Reference Page.

Ping sweep

A network probe technique that sends ICMP ECHO requests to multiple IP addresses 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 the context of network hardware, a connector for passing information into and out of a network device.

PostgreSQL

An open source relational database that NNMi uses by default to store information such as topology, incidents, and configuration information.

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.

Quick Start Configuration Wizard

A wizard that runs automatically immediately after installation of NNMi has been completed. This wizard enables you to configure community strings for reading an SNMPv1 or SNMPv2c environment, to set up discovery for a limited range of nodes, and to create an administrator account.

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.

regional manager

The NNMi management server in a global network management deployment that provides discovery, polling, and trap reception for devices, and that forwards information to the global manager.

role

See user role.

root cause analysis (RCA)

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 might result in status changes for managed objects, generation of root cause incidents, or both.

An 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 router failure that is the root cause has been 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*.

rule

See discovery rule.

rule-based discovery

Often called *auto-discovery*, NNMi can use rule-based discovery based on user-specified discovery rules to seek out nodes that NNMi needs to add to its database. NNMi looks for discovery hints in data from discovered nodes, and then checks those candidates against the specified discovery rules. You configure discovery rules in the **Discovery Configuration** area of the NNMi console under **Auto-Discovery Rule**. Contrast with *list-based discovery*.

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

A process, based on seeds or seed files, that discovers and returns detailed layer-2 connection information only about the nodes that you specify as seeds. Seeded discovery maintains a limited network inventory for specific queries and tasks. Contrast with *auto-discovery*. See also *spiral discovery*.

Simple Network Management Protocol (SNMP)

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 modified by remote users. SNMP is the predominant protocol used by NNMi to exchange network management information with agent processes on managed nodes. NNMi supports SNMPv1, SNMPv2, and SNMPv3, which are the three most common versions of SNMP.

SNMP

See Simple Network Management Protocol.

SNMP trap

Network management using polling (requests from an SNMP manager and 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 large number of unsolicited SNMP agent messages, which can overwhelm an SNMP manager process (in this case, NNMi). You can configure an SNMP trap storm threshold in NNMi, using the nnmtrapconfig.ovpl command. 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 account

In NNMi, a special account provided for use during NNMi installation. After installation, the NNMi system account is used only for command-line security and recovery purposes. Contrast with *user account*.

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.

topology (network)

In communication networks, a schematic description of the arrangement of a network, including its nodes and connections.

trap

See SNMP trap.

trap-receiving component

The portion of a Northbound application that receives SNMP traps.

Some applications include a separately installable component that receives SNMP traps and forwards them to another component for processing. For any Northbound application that does not include such a component, *trap-receiving component* is synonymous with *Northbound application*.

unconnected interface

From NNMi's perspective, an unconnected interface is an interface that is not connected to any other device discovered by NNMi. By default, the only unconnected interfaces that NNMi monitors are those that have IP addresses are contained in nodes from the **Routers** node group.

user account

A means in NNMi of providing a user or group of users with access to NNMi. 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 can access the NNMi console, as well as which workspaces and actions are available to each user account. NNMi provides hierarchical user roles, which are predefined by the program and cannot be modified. Such roles include the following:

• Administrator

- Web service client
- Operator level 2
- Operator level 1
- Guest

See also user account.

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

Α

abbreviations for products 631 account 634 acronyms 631 active 343 active cluster node 375, 634 active protocol 88 address hint 634 AddressNotResponding incident 588 administrator privilege 54 administrator role 62 agent 589 agent query not responding to 589 responding to 589 all scope 427 analyzing (managed node) 586 application failover 343, 634 behavior 353 cluster manager (mode) 353 configuring NNMi for 347 disabling 360 incident 357 network latency/bandwidth consideration 369 NNMi management server 344 NNMi database 369 setup 344 using feature of 353 Application_A.log file 423 approach list-based discovery 107 rule-based discoverv 108 architecture 374 ARP cache 111, 634 attribute author 71 ordering 73 authentication failure, reducing 103 authentication profile 89 author attribute 71 authoritative server 32 auto-discovery 634 configuring 65 auto-discovery rule 48, 65

auto-discovery rule ordering 108

В

backup all scope 427 configuration scope 427 database 434 event scope 427 of data before upgrading 433 427 offline online 427 strategy 432 topology scope 427 bandwidth consideration 369 benefit list-based discovery 107 rule-based discovery 108 best practice author attribute 71 checking order number 132 object group definition 126 ordering attribute 73 preparing NNMi configuration to be moved 485 reusable node group 127 saving existing configuration 70 short polling interval 128 Block SNMPv1 or SNMPv2c Traps 448

С

cache (ARP) 111 category, status 586 Causal Engine 584,634 causality 634 certificate Certificate Authority 62 self-signed 62 changing management 610 network 610 changing NNMi from HP-UX or Solaris to Linux 547 checking communication setting 102 management IP address 101

node configured for SNMP 101 node group 135 order number 132 SNMP access 101 checklist 123 Cisco 74 router 74 switch cluster 634 cluster architecture for HA 374 cluster manager 343 cluster member 343, 634 cluster node 634 active 375, 634 passive 375 cluster.log file 423 collecting data, validating (Status Poll) 136 com.hp.ov.nms.cluster.timeout.archive 353 command nnm.envvars.bat 582 nnm.envvars.sh 582 nnmbackup.ovpl 427 nnmcluster 347 nnmcommconf.ovpl 102 nnmconfigimport.ovpl 487 nnmdatareplicator.conf 402 nnmdatareplicator.ovpl 402 nnmhargconfigure.ovpl 417 nnmloadseeds.ovpl 114, 525 nnmsnmpwalk.ovpl 59 ovstart 58, 357 ovstop 58, 357 ping 589 command line mode 353 command line security 45 communication concept 82 configuration 91 configuration region 87 creating plan 87 evaluating setting 101 tuning 103 communication settings, configuring 439 community string 64, 634 comparison customizing event monitoring 511 network discovery 507

status monitoring 509 component health monitoring 123 concept Causal Engine 584 communication 82 configuration 68 discovery 507 event monitoring 511 HA 374 state polling 122 status monitoring 509 conclusion 634 configuration concept 68 configuring communication 91 configuring HA 381 DNS 32 evaluating state polling 135 HA troubleshooting 416 information (NNMi) 383 list-based discovery 111 log file 423 manpage 376 moving NNMi 487 node 88 polling example 123 preparing NNMi to be moved 485 Quick Start Configuration Wizard 34 region 87 resetting 80 rule-based discovery 111 saving existing 70 script (HA cluster) 422 SNMP access 516 state polling 122 updating on transaction basis 72 configuration file, replication of 402 configuration scope 427 Configuration workspace configuring state polling 132 evaluating state polling 135 Configure TLS Protocols 492 configuring community string 64 network discovery 64 connection operation (down) 594

operation (up) 594 router (down) 600 router (up) 600 ConnectionDown incident 588 console 634 containment (node group) 75 controller 634 conventions abbreviations for products 631 acronyms 631 fonts and symbols 8 KB, MB, GB, and TB 633 CPU resource 138 creating communication plan 87 object group definition 126 reusable node group 127 shadow 598 custom attributes, replicating from regional manager to global manager 322 customizing, event monitoring 511

D

daemon mode 353 data checking data collection 136 collecting (State Poller) 129 shared disk 401 data restoration, script 430 database moving 486 resetting 80 topology 122 database, resetting 80 default community string 103 discovery 105 router 119 rule-based discovery 114 setting 79 switch 119 default value environment variable (Windows) 582 default value (Linux) environment variable 582 definition 588

device excluding, from discovery 109 filter 76 trap, generated 610 device profile, customizing 520 device support, using NETCONF 92 DHCP 28 disabling SNMP 88 traffic 86 disadvantage list-based discovery 107 rule-based discovery 108 Discover and monitor Cisco ACI networks 97 Discover and monitor VMware hypervisor-based virtual networks 94 discovering link aggregation server-to-switch 115 discovery 64, 522 auto-discovery rule 65 checking progress 67 deleting node 117 disadvantage of 108 discovery configuration checklist 64 discovery mode 65 discovery seed 65 evaluating 117 excluding device from 109 hint 634 key concept of 507 migrating 522 performance 103 process 634 restarting 80 router 118 rule 634 seed 634 selecting primary approach 107 spiral 65, 104 switch 118 120 tuning discovery configuration checklist 64 discovery progress 67 discovery seed 48 disk copying data file (shared disk) 381 directory (shared disk) 401

failover 420 group (HA configuration) 383 disk group 383 DiskGroup A.log file 423 DNS servers, using multiple 32 DNS, checking for well-configured 32 Domain Name System 32 down administration (interface) 592 connection 594, 600 creating shadow (node) 598 distribution router (node) 595 595 node operation (interface) 591 router (node) 600 dynamic NAT 248 consideration on 259 discovery 261 global network management 262 hardware and software requirements for 261 subnet 262 dynamic PAT consideration on 259 discovery 261 global network management 262 hardware and software requirements for 261 subnet 262 dynamic port address translation (dynamic PAT) 248

Ε

embedded database tool, providing password 491 encryption, in NNMi 493 encryption, of user account passwords 493 end node (concept of discovery) 105 end-to-end diagnosis 584 environment variable 581 about 581 application failover 344 managing 582 MANPATH (Linux) 573 episode 585, 634 evaluating communication setting 101 configuring state polling 135 evaluation, order of 122 event 539 monitoring 511

event forwarding filter 558 event monitoring customizing 511 key concept of 511 event scope 427 example application failover 355 configuring node group 132 polling configuration 123 exclude NNMi files and directories from the scan target 45 extending monitoring 125

F

failover (disk) 420 failure network scenario 588 103 reducing authentication fault polling 634 file HA cluster 422 HA configuration 422 hostnolookup.conf 32 ipnolookup.conf 32, 519 netmon.cmstr 516 nms-auth-config.xml 242 nms-cluster.properties 347 nnmdatareplicator.conf 422 not updating on active node 417 nsswitch.conf 32 oid to sym 520 ov.conf 419 replication 402 replication for HA 402 snmpout.txt 516 system type 383 XML 487 file system type (HA configuration) 383 filter, device 76 filtering interface group 77 node group 74 firewall, disabling network access 86 flow model (task) 69 font conventions 8 form interface group 126

Interface Group Settings 138 Monitoring Configuration 122, 138 node group 126 Node Group Settings 138 fully qualified domain name 62 determining which 62 obtaining or setting official 35

G

GB meaning 633 generating, trap 610 global manager 634 global network management 634 group configuring 132 disk 383 filtering 77 interface (filtering) 77 preconfiguring 127 purpose 74 volume 383 quest role 62

Η

HA 634 HA cluster architecture 374 changing IP addresses 404 concept 374 file 422 NNMi 381 problem in starting 418 scenario 375 script 422 shared data 401 supported product 374 troubleshooting configuration 415 HA configuration 422 file 422 log file 423 maintaining 403 reference page 376 script 422 shared disk 383 HA information, NNMi 381 HA primary cluster node, configuration information 383

HA resource group 634 cannot start 417 configuring 383 description 374 stopping 411 HA nnmhaserver.log file 423 haconfigure.log file 423 hardware 27 information about 27 health information 136 hierarchy (node group) 75 high availability 634 cluster 373 host (virtual for HA configuration), NNMi 383 host name (changing for HA) 404 hostnolookup.conf file 32

I

ICMP 634 address monitoring 129 disabling traffic 86 IPv4 address 590 ICMP ping 81 incident 634 application failover 357 concept for 140 configuring 148 evaluating 153 example 588 planning 147 tuning 154 incident correlation notification 556 incident deletion notification 557 incident forwarding 555 incident lifecycle state change notification 556 Index Term 99 initial startup problem 58 installation problem 57 installing, preinstallation checklist 26 insufficient disk space 57 integrating NNMi with directory service 218 with directory service through LDAP 218 Integration with JP1/IM2 Intelligent Integrated Management Base 569 interactive mode 353

interface 634 administration 593 group 132 model 72 moving 610 operation 592 79 setting 586 status virtual host network in HA configuration 383 interface group 634 Interface Group Settings form 138 Interface Groups form 126 Interface Groups option 126 InterfaceDown incident 588 Internet Control Message Protocol 634 IP address changing for HA 404 entering discovery seed 48 management server 52 range 114 range of private 266 IP address range 48 valid 48 IP addresses managing overlapping, in NAT environment 245 mapping overlapping 266 ipnolookup.conf file 32, 519 IPv4 address 586 not responding to ICMP 590 responding to ICMP 590

J

JavaScript, enabling 62 JBoss port contention 58

Κ

KB meaning 633

L

L2 634 L3 634 latency (network) 83 Layer 2 634 Layer 3 634 LDAP configuration file 242 license 52 adding 117 checking type of 52 limitation 108 link aggregation, discovering 115 Linux, installing required library in 37 list-based discovery 634 overview 107 local Northbound application 562 log file (HA cluster), configuring 423 logical volume 375, 634

Μ

maintenance mode 403 managed node, analyzing 586 managed nodes, checking number of 52 management server 634 DHCP 28 IP address 52 management address preference 85 Management Information Base 634 management, configuration change 610 manpage 376 MB meaning 633 memory resource 138 MIB 634 MIB II variable 129 MIB read during new installation 574 migrating NNMi from HP-UX or Solaris operating system 546 migration procedure 514 MINCAUSE algorithm 584 model, user interface 72 monitoring 125 concept (event monitoring) 511 concept (status monitoring) 509 configuring (status monitoring) 509 customizing (event monitoring) 511 extending 125 node (network) 138 setting 79 Monitoring Configuration form 122, 126, 133 description 122 setting polling interval 126 setting polling type 126 tuning state polling 138 mount point 383

Mount_A.log file 423 moving interface 610 NNMi configuration 487 NNMi management server 484 Multihomed NNMi Management Server 99 multiple 89

Ν

name resolution, restricting 519 NAT 246 benefits of 247 implementing in NNMi 249 NAT type 248 NETCONF about 93 enabling and configuring in managed device 94 for device support, using 92 NETCONF device credential in NNMi, configuring 94 NETCONF protocol operation 93 netmask (virtual host in HA configuration) 383 netmon.cmstr file 516 network backbone 105 configuration change 609 connectivity 117 failure scenario 588 load 138 node 138 verifying connectivity 117 network discovery 65 network failure scenario 588 network interface (virtual host in HA configuration) 383 network latency 83 consideration 369 network monitoring, verifying configuration for 135 network operating center 517 network topology 65 nms-auth-config.xml file 242 nms-cluster.properties file 347 NmsApa service Causal Engine 584 configuration change 610 device-generated trap 610 network connection 610 setting status on object 585

nmsdbmgr service disk failover 420 problem in starting 419 NNM event 634 NNM version 8 or earlier comparison with 506 upgrading from 513 nnm.envvars.bat command 582 nnm.envvars.sh command 582 nnmbackup.ovpl 427 nnmcluster command 347 NnmClusterFailover incident 357 NnmClusterStartup incident 357 nnmcommconf.ovpl command 102 nnmconfigexport.ovpl, outputting configuration to XML 487 nnmconfigimport.ovpl command 487 nnmdatareplicator.conf file 422 nnmdatareplicator.ovpl script 422 nnmhaclusterinfo.ovpl script 422 nnmhaconfigure.ovpl script 422 nnmhadisk.ovpl command, troubleshooting nmsdbmgr 419 script 422 nnmhamonitor.ovpl script 422 nnmhamscs.vbs scrip 422 nnmharg.ovpl script 422 nnmhargconfigure.ovpl command 417 script 422 nnmhastart.ovpl script 422 nnmhastartrg.ovpl command 417 script 422 nnmhastop.ovpl script 422 nnmhastoprg.ovpl 422 nnmhaunconfigure.ovpl script 422 NNMi 634 accessing 62 backing up and restoring 363 implementing NAT in 249 in HA cluster, maintaining 404 installing 40 installing (Linux) 43 installing (Windows) 40 licensing 52 list of ports 612
modifying setting 366 moving database 486 removing 54 removing (Linux) 55 removing (Windows) 54 starting, stopping, and restarting 362 unconfiguring from HA cluster 408 upgrading (including applying patch) 362 with directory service, integrating 218 NNMi after HA maintenance restarting 405 starting 405 NNMi configuration moving 486 to be moved 486 to be moved, preparing 485 NNMi configuration, resetting 80 NNMi console 634 accessing 62 enabling Web browser 35 sign-in 62 updating on transaction basis 72 URL 62 NNMi database application failover 369 changing password 368 NNMi Help, accessing 63 NNMi installation, when disk space is insufficient 57 NNMi management server 28, 634 changing 488 changing domain name of 489 changing host name 489 changing IP address of stand-alone 488 upgrading 496 NNMi Northbound application connection parameter 563 NNMi Northbound interface 552, 553, 634 application failover 562 changing 559 destination status information 566 disabling 560 enabling 554 integration content 564 MIB information used by 566 overview 553 SNMP trap information used by 566 troubleshooting 561

using 555 NNMi Northbound Interface Destination form reference 563 NNMi Quick Start Configuration Wizard 34 NNMi service, restarting 58 NNMi spiral discovery 65 NNMi user access information 219 NNMi user group 221 NNMi, calculations for state and status 264 NNMi, deployment in network address translation (NAT) environment 262 NNMi, migrating to discovery 522 event 539 SNMP 516 status monitoring 531 nnmloadseeds.ovpl command 114, 525 nnmofficialfqdn.ovpl script 62 nnmrestore.ovpl script 430 nnmsnmpwalk.ovpl command 59 nnmtrapd.conf file 542 NOC 517 634 node clearing shadow 599 configuration 88 deleting 117 deleting discovered 117 distribution router 596 group 132 monitoring 138 reachable 600 router 600 settina 79 status 586 unreachable 599 updated 610 updating 610 node group 634 checking 135 configuration 132 defining 74 device filter 76 hierarchy 75 interface group 77 membership 75 non-SNMP device 125 preconfiguring 127

status 77, 586 node group map settings, configuring 438 Node Group Settings form 138 Node Groups form 126 Node Groups option 126 node groups, configuring 437 NodeDown incident 588 NodeOrConnectionDown incident 588 Non-SNMP Devices node group 125 Nortel router 74 switch 74 Northbound application 553, 634 Northbound destination 553, 634 not responding, IPv4 address to ICMP 590 nslookup request, avoiding 32 nslookup, improving response time for 32 nsswitch.conf file 32

0

object identifier 634 object (setting status) 585 object group definition 126 offline backup 427 OID 634 oid to sym file 520 ongoing discovery 65 online backup 427 operator level 1 62 operator level 2 62 option interface group 126 node group 126 order (evaluation) 122 order attribute, best practice 71 order number (checking) 132 Ordering attribute, auto-discovery rule 108 ov.conf file 419, 422 ov.conf, HA configuration 422 ovstart command 58, 357, 634 ovstatus command 634 ovstop command 58, 357, 634

Ρ

page 376 passive cluster node 375

password in directory service 221 password, sign-in 62 performance (status polling) 136 permanent license 52 permanent license key 52 obtaining and installing 52 preparing to install 52 personal computer (concept of discovery) 105 phase 514 ping command 589 request 129 ping node 59 Ping sweep 109, 634 place ipnolookup.conf 519 netmon.cmstr 516 planning polling interval 128 state polling 122 point (mount) 383 polling checklist 123 configuration example 123 evaluating performance 136 planning interval 128 protocol 86 starting 136 tuning (status) 138 port 634 JBoss port contention 58 ports, list of 612 Postgres 343 PostgreSQL 634 pre-configured user role 62 preconfiguring interface group 127 node group 127 preference, SNMP version 84 preinstallation checklist 26 NNMi management server 28 NNMi Quick Start Configuration Wizard 34 primary cluster node 375 printer (concept of discovery) 105 private IP address, range of 266 problem (starting HA) nmsdbmgr 419

NNMi 418 problem in starting nmsdbmgr 419 NNMi 418 profile (device), concept 76 protocol active 88 communication 82 polling 86 SNMP 586 public key certificate 634

Q

Quick Start Configuration Wizard 634 URL 48 using 48

R

range (IP address) 118 RCA 634 reachable node 600 recovery.conf file 353 reducing authentication failure 103 default community string 103 reference page 376 refining state polling 122 region 634 communication configuration region 87 regional manager 634 remote Northbound application 562 renaming host, NNMi 404 replication of configuration file 402 request (SNMP/ICMP request) 103 resource (system) 138 resource group 383 responding, IPv4 address to ICMP 590 restarting discovery 80 NNMi after HA maintenance 405 restoration database 434 file system file only 433 script 430 strategy 432

retry timeout 87 103 tuning role 62, 634 root cause 585 generating conclusion 584 root cause analysis 634 root cause incident 634 root privilege 55 round-robin DNS 32 router default 119 defining node group 74 discovery 118 hierarchy 75 monitoring 125 rule 634 ordering (auto-discovery) 108 rule-based discovery 634 overview 108

S

saving, existing configuration 70 scenario application failover 355 HA cluster 375 network failure 588 script HA configuration 422 nnmbackup.ovpl 426 nnmbackupembdb.ovpl 426 nnmhaclusterinfo.ovpl 422 NNMi data, restoring 430 nnmresetembdb.ovpl 426 nnmrestore.ovpl 426 nnmrestoreembdb.ovpl 426 script, nnmofficialfqdn.ovpl 62 secondary cluster node 375 secondary DNS service 32 seed 634 rule-based discovery 108 seeded discovery 634 server moving NNMi 485 service (NmsApa) configuration change 610 device-generated trap 610

network connection 610 status 585 updating node 610 service level agreement 128 setting, status on object 585 shadow clearing 599 creating 598 shared disk copying data file 381 401 data directory on 401 shared disk format 383 shared file system type (HA configuration) 383 shared HA data 401 sign-in 62 Simple Network Management Protocol (SNMP) 634 SNMP 81, 634 agent status 586 communication 88 communication problem 117 component health 129 configuring access 516 migrating to NNMi 516 monitoring 129 node configuration 101 protocol 586 request 103 supported version 64 tuning setting 103 version preference 84 SNMP information example of 516 migrating 516 SNMP proxy, configuring 91 SNMP trap 634 SNMP trap storm 634 SNMP traps determining which to send to NNMi 129 snmpout.txt file 516 SNMPv1 trap 255 SNMPv2c trap 253 SNMPv3 credentials 70 SNMPv3 informs 85 SNMPv3 traps 85 software 27 information about 27

Software License Agreement 52 spiral discovery 65, 104, 634 standby 343 state 634 State Poller communication setting 102 concept 122 configuration 122 evaluating configuration 135 health information 136 planning 122 symptom 584 tuning 138 What Can NNMi Monitor 124 state polling 634 refining 122 tuning 138 static NAT 248 251 communication using 250 consideration on discovery 252 global network management 258 subnet 258 trap 253 status 586, 634 interface 586 node 586 node group 586 object 585 SNMP agent 586 status monitoring 509 key concept 509 migrating 531 status poll, initiating 136 stopping NNMi (HA resource group) 411 NNMi (without causing HA failover) 405 strategy backup 432 restoration 432 supported version, SNMP 64 sweep 109 switch default 119 defining node group 74 discovery 118 hierarchy 75

troubleshooting cannot start NNMi console 59 discovery 59 does not open NNMi console 59 HA configuration 416 initial startup 58

Symantec Cluster Server

HA resource group

symbol conventions 8

resource 138

system account 62, 634

setting password 38

system object ID 634

system object ID range

auto-discovery

evaluating 119

TB meaning 633

temporary license

term, HA 375

timeout 83

tuning

value

topology

trap 634

database

network 634

topology scope

traffic, disabling

generating 610

tip

temporary license key 52

103

88

IP address range

system ID range

tip (for configuration)

auto-discovery 114

65

seeded discovery 114

122

427

86

trap-receiving component 553, 634

testing, configuring command 91

symbolic link

sysObjectID

system

Т

nnmharg.ovpl script 422

57

634

shared file type (HA configuration)

114

task, network discovery configuration

52

118

119

374

383

64

installation 57

installation and initial startup 57 NNMi-specific HA 418 tuning communication 103 discovery 120 state polling 138

U

unconnected interface 634 unreachable node 599 up administration 593 clearing shadow 599 distribution router 596 operation 592 updating, node 610 user 62 sign-in 62 user account 634 user interface model 72 user name in directory service 221 user role 634 pre-configured 62 system account 62

V

variable (MIB II) 129 verifying communication setting for device 102 network connectivity 117 Veritas Cluster Server HA resource group 374 nnmharg.ovpl script 422 version comparison customizing event monitoring 511 network discovery 507 status monitoring 509 version, SNMP preference 84 virtual host (HA configuration) netmask 383 network interface 383 virtual host name 383, 634 virtual IP address 634 volume group 375, 383, 634 Volume A.log file 423

W

Web browser accessing NNMi console 62 enabling 35 supported 28 Windows Server Failover Cluster HA resource group 374 nnmhamscs.vbs script 422 workspace, configuring state polling 132

Х

XML file 80, 487

@Hitachi, Ltd.

6-6, Marunouchi 1-chome, Chiyoda-ku, Tokyo, 100-8280 Japan