



Hewlett Packard
Enterprise

HPE Network Node Manager i Software

Software Version: 10.30
for the Windows® and Linux® operating systems

Deployment Reference

Document Release Date: June 2017
Software Release Date: June 2017

Legal Notices

Warranty

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

The information contained herein is subject to change without notice.

Restricted Rights Legend

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

Oracle Technology — Notice of Restricted Rights

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

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

Copyright Notice

© Copyright 2008–2017 Hewlett Packard Enterprise Development LP

Trademark Notices

Adobe® is a trademark of Adobe Systems Incorporated.

Apple is a trademark of Apple Computer, Inc., registered in the U.S. and other countries.

AMD is a trademark of Advanced Micro Devices, Inc.

Google™ is a registered trademark of Google Inc.

Intel®, Intel® Itanium®, Intel® Xeon®, and Itanium® are trademarks of Intel Corporation in the U.S. and other countries.

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

Internet Explorer, Lync, Microsoft, Windows, and Windows Server are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

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

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

sFlow is a registered trademark of InMon Corp.

UNIX® is a registered trademark of The Open Group.

Acknowledgements

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

This product includes software developed by the Visigoth Software Society (<http://www.visigoths.org/>).

Documentation Updates

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

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

To check for recent updates or to verify that you are using the most recent edition of a document, go to:
<https://softwaresupport.hpe.com/group/softwaresupport/search-result?keyword=>.

This site requires an HP Passport account. If you do not have one, click the **Create an account** button on the HP Passport Sign in page.

Support

Visit the HPE Software Support web site at: <https://softwaresupport.hpe.com>

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

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

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

Most of the support areas require that you register as an HP Passport user and sign in. Many also require a support contract. To register for an HP Passport ID, go to <https://softwaresupport.hpe.com> and click **Register**.

To find more information about access levels, go to:

<https://softwaresupport.hpe.com/web/softwaresupport/access-levels>

Contents

Chapter 1: About This Guide	17
What Is in This Guide?	17
Path Conventions Used in This Document	17
Revision History	18
For More Information about NNMi	19
Chapter 2: Preparation	21
Hardware and Software Requirements	21
Supported Hardware and Software	21
Checking for Required Patches	21
System Configuration (Linux)	22
Installing NNMi and the NNM iSPiS	22
NNMi Smart Plug-In Version Requirements	22
Chapter 3: Configuration	23
General Concepts for Configuration	24
Best Practice: Use the Author Attribute	25
User Interface Model	25
Ordering	25
Node Groups and Interface Groups	26
Group Overlap	26
Node Group Membership	27
Hierarchies/Containment	27
Device Filters	28
Additional Filters	28
Additional Nodes	28
Node Group Status	29
Interface Groups	29
Node Interface and Address Hierarchy	29
Reset the NNMi Configuration and Database	30
Configure NNMi to Use a Different Java Development Kit	31
NNMi Communications	35
Concepts for Communications	36
Levels of Communication Configuration	36
Network Latency and Timeouts	37
SNMP Access Control	37
SNMP Version Preferences	38
Management Address Preferences	39
SNMPv3 Traps and Informs	40
Polling Protocols	40
Communication Configuration and the <code>nnmsnmp*.ovpl</code> Commands	41
Plan Communications	41
Default Communication Settings	41
Communication Configuration Regions	41
Specific Node Configurations	42

Retry and Timeout Values	43
Active Protocols	43
Multiple Community Strings or Authentication Profiles	43
SNMPv1 and SNMPv2 Community Strings	44
SNMPv3 Authentication Profiles	44
Configure Communications	44
Configuring SNMP Proxy Settings	45
Device Support Using the Network Configuration Protocol (NETCONF)	46
What is Network Configuration Protocol (NETCONF)?	46
Network Configuration Protocol (NETCONF) Operations	47
Enabling and Configuring Network Configuration Protocol (NETCONF) in a Managed Device	47
Configuring Network Configuration Protocol (NETCONF) Device Credentials in NNMi	47
Configuring Communication for Virtual Environments	48
Prerequisites to Monitor Virtual Machines Hosted on Hypervisors	48
Replacing the VMware Default Certificate	49
Configuring NNMi to Communicate with Hypervisors Using HTTPS	50
Enable HTTP to Communicate with Hypervisors	52
Evaluate Communications	53
Are All Nodes Configured for SNMP?	53
Is SNMP Access Currently Available for a Device?	53
Is the Management IP Address for SNMP Devices Correct?	53
Is NNMi Using the Correct Communications Settings?	53
Do the State Poller Settings Agree with the Communication Settings?	54
Tune Communications	54
NNMi Discovery	55
Concepts for Discovery	56
NNMi Derives Attributes through Device Profiles	57
Plan Discovery	57
Select Your Primary Discovery Approach	58
List-Based Discovery	58
Rule-Based Discovery	58
Auto-Discovery Rules	59
Auto-Discovery Rule Ordering	59
Exclude Devices from Discovery	59
Ping Sweep	60
Discovery Seeds for Auto-Discovery Rules	60
Best Practices for Auto-Discovery Rules	60
Discovery Rule Overlap	61
Limit Device Type Discovery	61
Node Name Resolution	61
Subnet Connection Rules	62
Discovery Seeds	62
Rediscovery Interval	63
Do Not Discover Objects	63
Discover Interface Ranges	64
Monitor Virtual IP Addresses with NNMi	64
Use Discovery Hints from SNMP Traps	65

Configure Discovery	65
Tips for Configuring Auto-Discovery Rules	65
Tips for Configuring Seeds	65
Discovering Link Aggregation	66
Discovering Server-to-Switch Link Aggregations (S2SLA)	66
Evaluate Discovery	67
Follow the Progress of Initial Discovery	67
Were All Seeds Discovered?	68
Do All Nodes Have a Valid Device Profile?	68
Were All Nodes Discovered Properly?	68
Auto-Discovery Rules	69
IP Address Ranges	69
System Object ID Ranges	69
Are All Connections and VLANs Correct?	70
Evaluate Layer 2 Connectivity	70
NNMi Discovery and Duplicate MAC Addresses	70
Rediscover a Device	71
Tune Discovery	71
Discovery Log File	71
Unnumbered Interfaces	71
Controlling Deletion of Unresponsive Objects	72
NNMi State Polling	73
Concepts for State Polling	74
Plan State Polling	74
Polling Checklist	74
What Can NNMi Monitor?	76
Stop Monitoring	76
Interfaces to Unmonitored Nodes	77
Extend Monitoring	77
Planning Groups	78
Interface Groups	79
Node Groups	79
Planning Polling Intervals	80
Deciding What Data to Collect	81
Deciding What SNMP Traps to Send to NNMi	82
Configure State Polling	83
Configure Interface Groups and Node Groups	83
Configure Interface Monitoring	84
Configure Node Monitoring	84
Verify Default Settings	84
Evaluate State Polling	85
Verify the Configuration for Network Monitoring	85
Is the interface or node a member of the right group?	85
Which settings are being applied?	85
Which data is being collected?	86
Evaluate the Performance of Status Polling	86
Is the State Poller keeping up?	86
Tune State Polling	87

NNMi Incidents	89
Concepts for Incidents	90
Incident Lifecycle	90
Trap and Incident Forwarding	91
Comparison: Forwarding Third-Party SNMP Traps to Another Application	93
MIBs	93
Custom Incident Attributes	94
CIAs Added to Closed Management Event Incidents	94
Incident Reduction	95
Incident Suppression, Enrichment, and Dampening	96
Lifecycle Transition Actions	96
Plan Incidents	97
Which Device Traps Should NNMi Process?	97
Which Incidents Should NNMi Display?	97
How Should NNMi Respond to Incidents?	97
Should NNMi Forward Traps to Another Event Receiver?	98
Configure Incidents	98
Configuring Incident Suppression, Enrichment, and Dampening	98
Configuring Lifecycle Transition Actions	98
Configuring Trap Logs	99
Configuring Incident Logging	99
Configuring Trap Server Properties	99
Batch Load Incident Configurations	100
Generating an Incident Configuration File with nnmincidentcfgdump.ovpl	101
Loading Incident Configurations with nnmincidentcfgload.ovpl	101
Evaluate Incidents	102
Tune Incidents	102
Enabling and Configuring Incidents for Undefined Traps	103
Configure NNMi Console	104
Reduce the Maximum Number of Nodes Displayed in a Network Overview Map	105
Reduce the Number of Displayed Nodes on a Node Group Map	105
Configure Gauges in the Analysis Pane	106
Limit the Number of Gauges Displayed	106
Setting the Refresh Rate for Gauges in the Analysis Pane	106
Eliminate Gauges from the Display	107
Control the Order of Displayed Node Gauges	107
Control the Order of Displayed Interface Gauges	107
Control the Order of Displayed Custom Poller Gauges	107
Understand how Gauge Properties are Applied	108
Troubleshoot Gauge Problems	108
Too Many Gauges Are Displayed	108
Configuring Map Label Scale Size and Borders	108
Configuring Auto-Collapse Thresholds for Loom and Wheel Diagrams	109
Disable the Analysis Pane	110
Customize Device Profile Icons	110
Configure a Table View's Refresh Rate	110
NNMi Auditing	112
Disable Auditing	116

Specify the Number of Days to Retain NNMi Audit Logs	116
Configure the Actions Included in the NNMi Audit Log File	117
About the NNMi Audit Log File	118
Chapter 4: Resilience	120
Configuring NNMi for Application Failover	122
NNMi Application Failover Overview	123
Application Failover Requirements	123
Set Up NNMi for Application Failover	124
Configuring your Cluster with the NNMi Cluster Setup Wizard (Embedded Database Users only)	127
Setting Cluster Communications (Optional)	128
Using the Application Failover Feature	128
Application Failover Behavior Using the Embedded Database	129
Application Failover Behavior Using an Oracle Database	130
Application Failover Scenarios	131
Additional ovstart and ovstop Options	132
Application Failover Incidents	132
Returning to the Original Configuration Following a Failover	133
NNM iSPIs and Application Failover	133
NNM iSPI Installation Information for Embedded Database	133
Deploying NNM iSPI in an Existing Application Failover Environment - Embedded Database	133
Deploying NNM iSPI with NNMi and then Configuring Application Failover - Embedded Database	135
NNM iSPI Installation Information for Oracle Database	135
Deploying NNM iSPIs in an Existing NNMi Application Failover Environment - Oracle Database	135
Deploying NNM iSPIs with NNMi and then Configuring Application Failover - Oracle Database	137
Integrated Applications	137
Disabling Application Failover	138
Administrative Tasks and Application Failover	140
Restoring NNMi Failover Environment	140
Application Failover and NNMi Patches	140
Applying Patches for Application Failover (Shut Down Both Active and Standby)	141
Applying Patches for Application Failover (Keep One Active NNMi Management Server)	142
Application Failover and Restarting the NNMi Management Servers	144
Application Failover Control after a Communication Failure	145
Application Failover and Recovery from a Previous Database Backup (Embedded Database Only)	145
Cluster File Transfer Warning Configurations	145
Network Latency/Bandwidth Considerations	146
Application Failover and the NNMi Embedded Database	147
Network Traffic in and Application Failover Environment	148
An Application Failover Traffic Test	148
Configuring NNMi in a High Availability Cluster	150
High Availability Concepts	151

High Availability Terms	152
NNMi High Availability Cluster Scenarios	153
Manpages	157
Verifying the Prerequisites to Configuring NNMi for High Availability	157
Configure High Availability	159
Configure NNMi Certificates for High Availability	159
Configure NNMi for High Availability	159
NNMi High Availability Configuration Information	163
Configuring NNMi on the Primary Cluster Node	165
Configuring NNMi on the Secondary Cluster Nodes	168
Configure NNM iSPIs for High Availability	169
NNM iSPI Performance for Metrics, NNM iSPI Performance for QA, and NNM iSPI Performance for Traffic	169
NNM iSPI Performance for QA, NNM iSPI for MPLS, NNM iSPI for IP Multicast, and NNM iSPI for IP Telephony	170
NNM iSPI Network Engineering Toolset Software and NNMi Running under HA	170
Configure NNMi for High Availability in an Oracle Environment	171
NNMi Dependency on Oracle in High Availability Environments	171
Configuring NNMi for High Availability in an Oracle Environment	171
Shared NNMi Data in High Availability Environments	172
Data on the NNMi Shared Disk in High Availability Environments	172
Replication of Configuration Files in High Availability Environments	173
Disabling Data Replication	173
Prepare the Shared Disk Manually in High Availability Environments	174
Configuring a SAN or a Physically Connected Disk	174
Setting the High Availability Variables in the ov.conf File	175
Moving the Shared Disk into the NNMiHA Resource Group	175
A Note about Shared Disk Configuration on Windows Server	175
Licensing NNMi in a High Availability Cluster	176
Maintaining the High Availability Configuration	177
Maintenance Mode	177
Putting an HA Resource Group into Maintenance Mode	177
Removing an HA Resource Group from Maintenance Mode	177
Maintaining NNMi in an HA Cluster	178
Starting and Stopping NNMi	178
Changing NNMi Hostnames and IP Addresses in a Cluster Environment	178
Stopping NNMi Without Causing Failover	180
Restarting NNMi after Maintenance	181
Maintaining Add-on NNM iSPIs in an NNMi HA Cluster	181
Unconfiguring NNMi from an HA Cluster	181
Running NNMi Outside HA with the Existing Database	183
Patching NNMi under HA	184
Troubleshooting the HA Configuration	185
Common High Availability Configuration Mistakes	185
Configuration Issues with RHCS 6	186
HA Resource Testing	186
Re-Enable NNMi for High Availability after All Cluster Nodes are Unconfigured	187
NNMi Does Not Start Correctly Under High Availability	188

Changes to NNMi Data are Not Seen after Failover	188
nmsdbmgr Does Not Start after High Availability Configuration	189
NNMi Runs Correctly on Only One High Availability Cluster Node (Windows)	190
Disk Failover Does Not Occur	190
Shared Disk is Not Accessible (Windows)	190
Shared Disk Does Not Contain Current Data	190
Shared Disk Files Are Not Found by the Secondary Node after Failover	190
Error: Wrong Number of Arguments	191
Resource Hosting Subsystem Process Stops Unexpectedly (Windows Server)	191
Product Startup Times Out (Windows WSCS 2008)	192
Log Files on the Active Cluster Node Are Not Updating	192
Cannot Start the NNMi HA Resource Group on a Particular Cluster Node	192
High Availability Configuration Reference	193
NNMi High Availability Configuration Files	193
NNMi-Provided HA Configuration Scripts	194
NNMi High Availability Configuration Log Files	195
Chapter 5: Modify Default Settings	197
Modify Access Control Lists for NNMi Folders	197
Change the Custom Poller Collections Export Directory	197
Changing the Maximum Amount of Disk Space for Custom Poller Collections Export	198
Configure Incident Actions	198
Overriding Settings in the server.properties File	201
Modify User Interface Properties	204
Configure the Data Payload Size in an ICMP Echo Request Packet	209
Configure how NNMi Determines the Host Name for a Device	211
Configure Character Set Encoding Settings for NNMi	211
Configure the Time NNMi Waits for an NNM iSPI Licensing Request	212
Modify NNMi Normalization Properties	213
Suppressing the Use of Discovery Protocols for Specific Nodes	214
Suppressing the Use of Discovery Protocol Collections	214
Suppressing the Monitoring of IP Addresses on Administrative Down Interfaces	215
Suppressing the Use of VLAN-indexing for Large Switches	216
Suppressing the Use of VLAN-indexing	216
Configuring Sensor Status	217
Configuring Physical Sensor Status	217
Propagating Physical Sensor Status to a Physical Component	218
Configuring Physical Sensor Status to not Propagate to the Physical Component	218
Overriding Physical Sensor Status Values	219
Configuring Node Sensor Status	219
Propagating Node Sensor Status to a Node	219
Configuring a Node Sensor's Status to not Propagate to the Node	220
Overriding Node Component Status Values	220
Chapter 6: Maintaining NNMi	222
NNMi Backup and Restore Tools	222
Backup and Restore Commands	222
Backing up NNMi Data	223
Backup Type	223
Backup Scope	223

Restoring NNMi Data	226
Same System Restore	227
Different System Restore	227
Restore on an NNMi Management Server Upgraded to 10.30	227
Restore in an HA Cluster	228
Backup and Restore Strategies	228
Back up All Data Periodically	228
Back up Data Before Changing the Configuration	229
Back up Data Before Upgrading NNMi or the Operating System	229
Restore File System Files Only	229
Backing up and Restoring the Embedded Database Only	229
Using Backup and Restore Tools in a High Availability (HA) Environment	230
Best Practices for Backup in an HA Environment	230
Best Practices for Restore in an HA Environment	230
NNMi Logging	230
NNMi Log Files	231
Sign-in and Sign-out Logging	231
Changing the Management Server	232
Best Practices for Preparing the NNMi Configuration to be Moved	232
Moving the NNMi Configuration and Embedded Database	233
Moving the NNMi Configuration	233
Restoring the NNMi Public Key Certificate	234
Task 1: Determine the Status of KeyManager Service	234
Task 2: Back up the Current nnm.keystore File	234
Task 3: Attempt to Locate the Original nnm.keystore File	235
Task 4: If Available, Restore the Original nnm.keystore File	236
Changing the IP Address of a Standalone NNMi Management Server	236
Changing the Hostname or Domain Name of an NNMi Management Server	237
Changing the Oracle Database Instance Connection Information	237
Task 1: Update the Oracle Database Instance	238
Task 2: Update the NNMi Configuration	238
Changing the Password that NNMi uses to Connect to the Oracle Database Instance	239
Modifying the Embedded Database Port	239
Schedule Outages	240
NNMi Self Monitoring	240
Chapter 7: Advanced Configuration	241
Apply Licenses	241
Preparing to Install a Permanent License Key	241
Checking the License Type and the Number of Managed Nodes	242
Obtaining and Installing a Permanent License Key	242
Obtaining Additional License Keys	243
Managing Certificates	243
About NNMi Certificates	244
Configure an Upgraded NNMi Environment to Use the New Keystore	245
Using Certificates with the PKCS #12 Repository	248
Generating a Self-Signed Certificate	248
Generating a CA-Signed Certificate	249
Types of CA-Signed Certificates	253

Delete a Certificate from the NNMi Keystore	255
Replacing an Existing Certificate with a new Self-Signed or CA-Signed Certificate	256
Working with Certificates in Application Failover Environments	257
Working with Certificates in High-Availability Environments	258
Configuring High-Availability Using Default Certificates	259
Configuring High-Availability Using New Certificates	259
Working with Certificates in Global Network Management Environments	259
Configuring Certificates in Global Network Management Environments	260
Configuring Certificates in Global Network Management Environments with Failover	261
Configuring an SSL Connection to the Directory Service	262
Using Certificates with the JKS Repository	265
About NNMi JKS Certificates	266
Replacing an Existing Certificate with a New Self-Signed or CA-Signed Certificate	267
Generating a Self-Signed Certificate	267
Generating a CA-Signed Certificate	268
Types of CA-Signed Certificates	271
Working with Certificates in Application Failover Environments	273
Working with Certificates in High-Availability Environments	275
Configuring High-Availability Using Default Certificates	275
Configuring High-Availability Using New Certificates	275
Working with Certificates in Global Network Management Environments	276
Configuring Certificates in Global Network Management Environments	276
Configuring Certificates in Global Network Management Environments with Failover	277
Configuring an SSL Connection to the Directory Service	278
Using Single Sign-On (SSO) with NNMi	280
SSO Access for NNMi	281
Enabling SSO for a Single Domain	282
Enabling SSO for NNMi Management Servers Located in Different Domains	282
SSO Access for NNMi and the NNM iSPiS	283
Disabling SSO	284
SSO Security Notes	285
Configuring NNMi to Support Public Key Infrastructure User Authentication	286
User Authentication Strategies	287
Configuring NNMi for PKI User Authentication (X.509 Certificate Authentication)	287
Logging on to NNMi using a Client Certificate	290
Revoking Access for a User Having a Client Certificate	291
Special Considerations When PKI User Authentication in Global Network Management Environments	291
Certificate Validation (CRL and OCSP)	291
General Configuration for Certificate Validation Protocols	292
Configuring Protocol Order	292
Configuring Protocol Requests	292
Validating Certificates Using CRLs	293
Enabling and Disabling CRL Checking	293
Changing the CRL Enforcement Mode	294
Changing How Often a CRL Should be Refreshed	294
Changing the Maximum Idle Time for a CRL	295
CRL Expiration Warnings	295

Changing the Location for a CRL	296
Validating Certificates Using Online Certificate Status Protocol (OCSP)	296
Enabling and Disabling OCSP Checking	297
Changing the OCSP Enforcement Mode	297
Enabling Nonce	298
Specifying the URL of the OCSP Responder	299
Configuring NNMi to Restrict Certificates Used for NNMi Logon Access	299
Example: Configuring NNMi to Require a Smart Card Logon	300
Configuring CLI Authentication for PKI User Authentication	303
Setting ACLs to Enable Non-Root Users to Run CLI Commands	304
Troubleshooting PKI User Authentication Issues	305
Configuring the Telnet and SSH Protocols for Use by NNMi	306
Disable the Telnet or SSH Menu Item	307
Configure a Telnet or SSH Client for the Browser	307
Integrating NNMi with a Directory Service through LDAP	308
NNMi User Access Information and Configuration Options	308
Internal Mode (Originally Referred to as Option 1): All NNMi User Information in the NNMi Database	310
Mixed Mode (Originally Referred to as Option 2): Some NNMi User Information in the NNMi Database and Some NNMi User Information in the Directory Service	310
External Mode (Originally Referred to as Option 3): All NNMi User Information in the Directory Service	311
Configuring NNMi to Access a Directory Service	312
Directory Service Queries	323
Directory Service Access	324
Directory Service Content	324
Information Owned by the Directory Service Administrator	328
User Identification	330
User Group Identification	331
Configuring User Group Retrieval from the Directory Service (Detailed Approach)	332
Directory Service Configuration for Storing NNMi User Groups	334
Verify the Directory Service Configuration	334
LDAP Configuration File Reference	335
nms-auth-config.xml	335
ldap.properties	338
Examples	342
Switching to the nms-auth-config.xml File	343
Multihomed NNMi Management Server	343
Managing Overlapping IP Addresses in NAT Environments	344
What is NAT?	344
What are the Benefits of NAT?	345
What Types of NAT are Supported?	345
How is NAT Implemented in NNMi?	345
Static NAT Considerations	346
Hardware and Software Requirements and Static NAT	347
Overlapping IP Address Mapping	347
Private IP Address Ranges	348
Communication and Static NAT	348

Administering ICMP Polling of the Management Address in a Static NAT Environment	348
Enabling ICMP Polling of the Management Address in a NAT Environment	348
Discovery and Static NAT	349
Monitoring Configuration for Static NAT	350
Traps and Static NAT	350
SNMPv2c Traps	350
SNMPv1 Traps	352
Subnets and Static NAT	353
Global Network Management: Optional for Static NAT	354
Dynamic NAT and PAT Considerations	354
Hardware and Software Requirements and Dynamic NAT and PAT	356
Discovery Configuration for Dynamic NAT and PAT	356
Monitoring Configuration for Dynamic NAT	356
Subnets and Dynamic NAT and PAT	356
Global Network Management: Required for Dynamic NAT and PAT	357
Deploy NNMi in a Network Address Translation (NAT) Environment	357
NNMi Calculations for State and Status	359
NNMi Security and Multi-Tenancy	360
Effects of Limiting Object Access	361
The NNMi Security Model	362
Security Groups	363
Example Security Group Structure	364
The NNMi Tenant Model	367
Tenants	367
Example Tenant Structure	368
NNMi Security and Multi-Tenancy Configuration	370
Configuration Tools	371
Configuring Tenants	374
Configuring Security Groups	375
Verifying the Configuration	376
Exporting the NNMi Security and Multi-Tenancy Configuration	378
NNMi Security, Multi-Tenancy, and Global Network Management (GNM)	378
Initial GNM Configuration	379
GNM Maintenance	380
Including Select Interfaces in NPS Reports	381
Configuring Single Sign-On for Global Network Management	381
Configuring Forwarding Filters on the Regional Managers	384
Configuring a Forwarding Filter to Limit Forwarded Nodes	384
Connecting a Global Manager with a Regional Manager	385
Determining the Connection States from global1 to regional1 and regional2	386
Reviewing global1 Inventory	386
Disconnecting Communication between global1 and regional1	386
Discovery and Data Synchronization	387
Replicating Custom Attributes from a Regional Manager to the Global Manager	388
Status Poll or Configuration Poll a Device	388
Determining Device Status and NNMi Incident Generation using a Global Manager	390
Configuring Application Failover for Global Network Management	390

Verify the Global Network Management Configuration	391
Check Clock Synchronization Status	391
View System Information	391
Synchronize Regional Manager Discovery from a Global Manager	392
Recover a Destroyed Database on global1	393
Global Network Management and NNM iSPIs or Third-Party Integrations	393
Global Network Management and Address Translation Protocols	393
Configuring NNMi Advanced for IPv6	393
Feature Description	394
Prerequisites	395
Licensing	396
Supported Configuration	396
Management Server	396
Supported SNMP MIBs for IPv6	397
Installing NNMi	397
Deactivating IPv6 Features	398
IPv6 Monitoring Following Deactivation	399
IPv6 Inventory Following Deactivation	399
Known Issues When Cleaning Up IPv6 Inventory	399
Reactivating IPv6 Features	399
Chapter 8: NNMi Security	403
Configuring SSL Communications for Web Access and RMI Communications	403
Requirement for New NNMi 10.30 Installations	403
Allowing Non-Root Linux Users to Start and Stop NNMi	404
Providing a Password for Embedded Database Tools	404
Configuring NNMi to Enable or Disable SSLv3 Ciphers	405
Configuring NNMi Ciphers	407
NNMi Data Encryption	407
Encryption Configuration Files	407
Text Blocks in the Crypto Configuration Files	408
Encryption and Application Failover	408
Encryption and User Account Passwords	409
Chapter 9: Use HPE Operations Bridge Reporter to View Reports	411
Prerequisites	411
Configure NNMi to Export Data to OBR	411
Configure OBR to Use the Data Collected by NNMi	412
Use Reports	412
Chapter 10: Migrating Performance Insight (OVPI) SNMP Collections of Custom Report Packs to NNMi	413
Chapter 11: Use Case: SNMP v1 or v2c Management Through Net-SNMP Proxy	416
SNMP Get Requests through Net-SNMP Proxy	416
Configuring Net-SNMP Proxy Settings Using the Command Line Interface	417
Configuring Net-SNMP Proxy Settings Using the Graphical User Interface	419
Net-SNMP Trap Forwarding through NNMi	424
Appendix A: Additional Information	427
Manually Configuring NNMi for Application Failover	427
NNMi Environment Variables	430
Environment Variables Used in This Document	430

Other Available Environment Variables	431
NNMi and NNM iSPI Default Ports	433
HPE Network Node Manager i Software Ports	434
NNM iSPI for MPLS Ports	445
NNM iSPI for IP Telephony Ports	448
NNM iSPI for IP Multicast Ports	451
NNM iSPI Performance for Traffic Ports	454
NNM iSPI Performance for QA Ports	461
NNM iSPI Performance for Metrics and NPS Ports	465
NNM iSPI NET Ports	466
NNMi Configuration Issues	467
Send Documentation Feedback	469

Chapter 1: About This Guide



(1) First installation
or test bed

Follow steps in
NNMi Installation Guide



(2) Production deployment and
migration from previous versions

Read NNMi Deployment
Reference (this book)



This chapter contains the following topics:

- ["What Is in This Guide?" below](#)
- ["Path Conventions Used in This Document" below](#)
- ["Revision History" on the next page](#)
- ["For More Information about NNMi" on page 19](#)

What Is in This Guide?

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

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

Path Conventions Used in This Document

This document primarily uses the following two NNMi environment variables to reference file and directory locations. This list shows the default values. Actual values depend on the selections that you made during NNMi installation.

- *Windows Server*:
 - %NnmInstallDir%: <drive>\Program Files (x86)\HP\HP BTO Software
 - %NnmDataDir%: <drive>\ProgramData\HP\HP BTO Software

On Windows systems, note the following:

- The NNMi installation process creates these system environment variables, so they are always available to all users.
 - Use quotes whenever a path name includes spaces (for 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.

Additionally, this document references some of the NNMi environment variables that you can source as part of your user log-on configuration on the NNMi management server. These variables are of the form NNM_*. For information about this extended list of NNMi environment variables, see ["Other Available Environment Variables" on page 431](#).

Revision History

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

Document Release Date	Description of Major Changes
May 2014 (10.00)	Initial release.
December 2014 (10.01)	<p>Added Configuring the Locale to Use for Sort Order to the "Configure Incidents" chapter.</p> <p>Added Configuring NNMi to Enable or Disable SSLv3 Ciphers to the "NNMi Security" chapter.</p> <p>Updates to the Working with Certificates for NNMi information in the "Advanced Configuration" chapter.</p>
November 2015 (10.10)	<p>Added Configuring Communication for Virtual Environments to the "NNMi Communications" chapter.</p> <p>Removed Changing the Directory Service Access Configuration to Support the NNMi Security Model from the "Integrating NNMi with a Directory Service Through LDAP" chapter.</p>

For More Information about NNMi

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

What do you want to do?	Where to find more information
View a list of available documentation for this version of NNMi.	Download the <i>NNMi Documentation List</i> . Use this file to track additions to and revisions within the NNMi documentation set for this version of NNMi. Click a link to access a document on the HPE manuals web site.
Install NNMi, NNMi Advanced, NNMi Premium, or NNMi Ultimate (first time).	Download the <i>HPE Network Node Manager i Software Interactive Installation Guide</i> . This guide contains basic steps to install and un-install the product, plus how to do an initial configuration using the NNMi Quick Start Configuration Wizard.
Plan for network deployment, including links to system requirements.	See " Preparation " on page 21 of this guide.
Configure NNMi for a production environment.	See " Configuration " on page 23 of this guide.
Review NNMi configuration considerations for VMware Hypervisor-Based Virtual Networks.	See "Discovering and Monitoring VMware Hypervisor-Based Virtual Networks (NNMi Advanced)" and "Managing VMware Hypervisor-Based Virtual Networks (NNMi Advanced)" in the Help for Administrators.
Configure NNMi behind the scenes.	See " Advanced Configuration " on page 241 of this guide.
Maintain the NNMi configuration.	See " Maintaining NNMi " on page 222 of this guide.
Upgrade to NNMi from previous versions of Network Node Manager i Software.	See the <i>HPE Network Node Manager i Software Interactive Installation Guide</i> , available on the HPE manuals web site.
Reference NNMi environment variables, ports, and messages.	See " Additional Information " on page 427 of this guide.
Obtain more information about a specific topic.	Download by example documents and white papers. For a list of the white papers available, see the <i>NNMi Documentation List</i> .
Print the NNMi help.	Download PDFs of the help content. For a list of the help PDFs available, see the <i>NNMi Documentation List</i> .
Install the NNM iSPI NET (NNM iSPI NET) Diagnostics Server and learn about NNM iSPI NET functionality.	Download the <i>HPE NNM iSPI Network Engineering Toolset Planning and Installation Guide</i> from the Network Node Manager SPI for NET product category for the Windows operating system.
	<p>Note: The NNM iSPI NET Diagnostics Server requires</p>

What do you want to do?	Where to find more information
	<p>an NNM iSPI NET or NNMi Ultimate license. See the <i>HPE NNM iSPI Network Engineering Toolset Software Interactive Installation and Upgrade Guide</i> for information about how to install and configure this server.</p>
Obtain documentation about the NNMi Developer Toolkit (SDK).	See " Apply Licenses " on page 241 to review information related to the SDK, obtaining and installing an SDK license, and viewing SDK documentation and samples.

Chapter 2: Preparation

This section contains the following chapter:

- "Hardware and Software Requirements" below

Hardware and Software Requirements

This section contains the following topics:

Supported Hardware and Software

Before installing NNMi, read the information about NNMi hardware and software requirements described in the following table.

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

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

Software and Hardware Pre-Installation Checklist

Complete (y/n)	Document to Read
	<i>HPE Network Node Manager i Software Interactive Installation Guide</i> <ul style="list-style-type: none">• Filename = <i>nnmi_interactive_installation_en.zip</i>• Instructions Filename: <i>NNMiReadMeFirst_en.pdf</i>
	<i>NNMi Release Notes</i> <ul style="list-style-type: none">• Filename = <i>release_notes_nnmi_en.pdf</i>• NNMi console = Help > NNMi Documentation Library > Release Notes
	<i>NNMi Support Matrix</i> <ul style="list-style-type: none">• Filename = <i>support_matrix_nnmi_en.pdf</i>• NNMi console = Linked from the release notes

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

Checking for Required Patches

Before installing NNMi, check the NNMi Release Notes for any required operating system updates.

System Configuration (Linux)

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

Installing NNMi and the NNM iSPIs

If you plan to use any of the HPE NNM iSPIs along with NNMi, you must install NNMi before installing any of the HPE NNM iSPIs.

NNM i Smart Plug-In Version Requirements

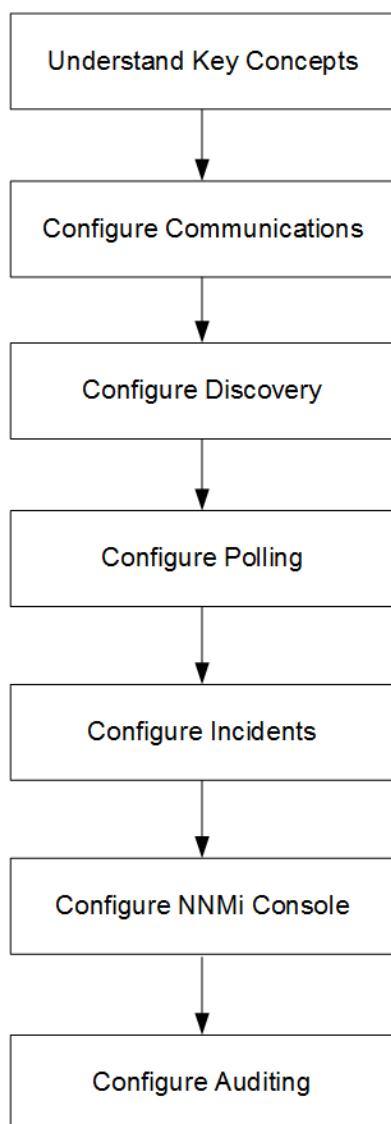
NNMi and each NNM i Smart Plug-In must have equivalent versions. For example, NNM iSPI Performance for Metrics version 10.10 is only supported with NNMi 10.10.

For a list of the iSPIs that are included with NNMi Premium and NNMi Ultimate, see the NNMi Release Notes, available at <http://h20230.www2.hp.com/selfsolve/manuals>.

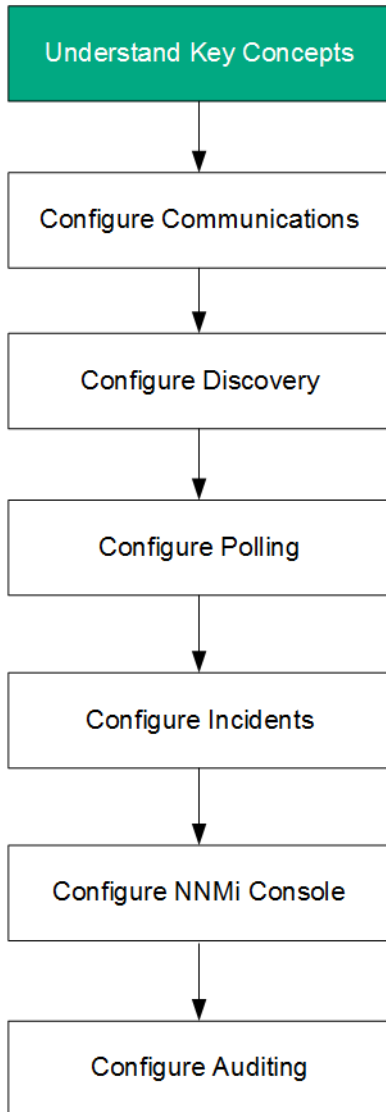
Chapter 3: Configuration

This section contains the following topics:

- "General Concepts for Configuration" on the next page
- "NNMi Communications" on page 35
- "NNMi Discovery" on page 55
- "NNMi State Polling" on page 73
- "NNMi Incidents" on page 89
- "Configure NNMi Console" on page 104
- "NNMi Auditing" on page 112



General Concepts for Configuration



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

This chapter contains the following topics:

- ["Best Practice: Use the Author Attribute" on the next page](#)
- ["User Interface Model" on the next page](#)
- ["Ordering" on the next page](#)
- ["Node Groups and Interface Groups" on page 26](#)
- ["Node Interface and Address Hierarchy" on page 29](#)
- ["Reset the NNMi Configuration and Database" on page 30](#)

Best Practice: Use the Author Attribute

Many NNMi configuration forms include the **Author** attribute.

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

When you upgrade NNMi, the installer does not overwrite any configurations whose author value is not HP.

User Interface Model

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 all of the way back to the NNMi console. If you close a form that contains unsaved changes (on that form or on a contained form), NNMi warns you about the unsaved changes and gives you a chance to cancel the close.

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

Ordering

Some NNMi console configuration forms include the **Ordering** attribute, which sets the priority for applying the configurations. For one configuration area, NNMi evaluates each item against the configurations from the smallest (lowest) ordering number to the next lowest ordering number, and so on, until NNMi finds a match. At that point, NNMi uses the information from the matching configuration and ceases to look for any more matches. (The communication configuration is an exception. NNMi continues to search for information 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 applies within the local context. The Menus and Menu Items tables contain multiple objects with the same ordering number because of the local context idea.

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

- Ordering on the **Menu** and **Menu Item** forms sets the order of items in the local context of the associated menu.
- Topology maps ordering on the **Node Group Map Settings** form sets the order of items in the **Topology Maps** workspace.

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

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

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

Node Groups and Interface Groups

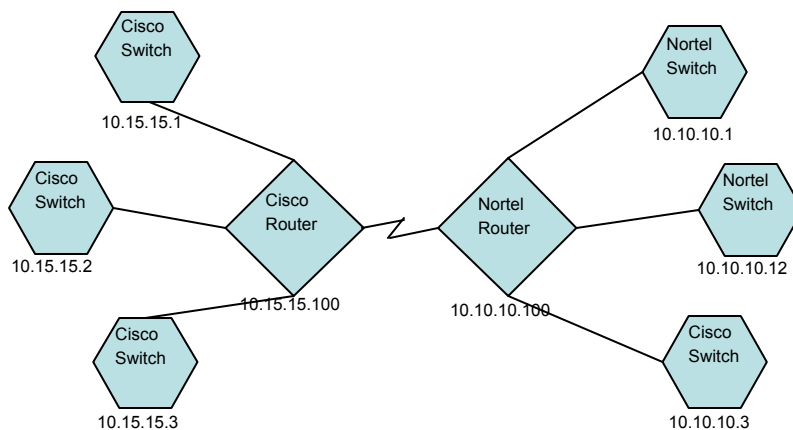
In NNMi, the primary filtering technique is to group nodes or interfaces, and then applying settings to a group or filtering visualizations by group.

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

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:

Node Group Overlap



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

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

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

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.

Caution: Rarely use the **Additional Nodes** tab to add nodes to a node group, as it 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.

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.

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.

The configuration interface automatically prevents circular hierarchy definitions.

Device Filters

During discovery, NNMi collects direct information through SNMP queries and derives other information from that through device profiles. (For more information, see ["NNMi Derives Attributes through Device Profiles" on page 57.](#)) By gathering the system object ID, NNMi can index through the correct device profile to derive the following information:

- Vendor
- Device category
- Device family within the category

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

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

Additional Filters

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

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

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

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

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

Additional Nodes

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 hostname. Only add nodes to a node group by individual hostnames as a last resort.

Caution: Rarely use the **Additional Nodes** tab to add nodes to a node group, as it consumes excessive resources on the NNMi management server.

Node Group Status

When configured to do so, 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, node group status calculation is off by default 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.

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.

Note: Interfaces in an Interface Group are not always initially excluded during discovery under the following conditions:

- The interface group is 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, it will be excluded when the exclusion filter is re-applied during a rediscovery.

See the NNMi Online Help for Administrators for more information about the Interface Capabilities provided by NNMi and the **Excluded Interfaces** Discovery Configuration option.

Node Interface and Address Hierarchy

NNMi assigns monitoring settings in the following manner:

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

based on the first matching **Node Settings** definition. The first match is the **Node Settings** definition with the lowest ordering number.

Note: 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.

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

Follow these steps:

1. Stop the NNMi services:

```
ovstop -c
```

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

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

3. Optional. If you want to keep any of the current NNMi configuration, use the `nnmconfigexport.ovpl` command to output the NNMi configuration to an XML file.

Tip: The `nnmconfigexport.ovpl` command does not retain SNMPv3 credentials. See the `nnmconfigexport.ovpl` reference page, or the Linux manpage, for more information.

4. Optional. Use the `nnmtrimincidents.ovpl` command to archive the NNMi incidents. Incidents are archived in the CSV format, as described in the `nnmtrimincidents.ovpl` reference page or Linux manpage.

5. Drop and recreate the NNMi database.

- For the embedded database, run the following command:

```
nnmresetembdb.ovpl -nostart
```

- For an Oracle database, ask the Oracle database administrator to drop and recreate the NNMi database. Maintain the database instance name.

6. If you have installed iSPi or stand-alone products that integrate with NNMi, reset those products to remove the old topology identifiers. For specific procedures, see the product documentation.

7. Start the NNMi services:

```
ovstart -c
```

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

8. Start configuring NNMi. Do one of the following:
 - Use the Quick Start Configuration Wizard.
 - Enter information into the **Configuration** workspace in the NNMi console.
 - Use the `nnmconfigimport.ovpl` command to import some or all of the NNMi configuration that you saved in [step 3](#).

Tip: If you are using the `nnmconfigimport.ovpl` command to import large amounts of configurations (such as 9,500 node groups or 10,000 incident configurations), consider using the `-timeout` option to adjust the import transaction timeout from its default value of 60 minutes (3600 seconds) to something longer. See the `nnmconfigimport.ovpl` reference page, or the Linux manpage, for more information.

Configure NNMi to Use a Different Java Development Kit

The NNMi installer, by default, installs OpenJDK 1.8. After installation, you can configure NNMi to use a different, non-default instance of Java Development Kit (JDK) 1.8 that is already installed on the NNMi management server.

Change JDK on a standalone NNMi management server

1. Log on to NNMi as root or administrator.
2. Run the following command:
 - *On Windows:* `%nnminstalldir%\bin\nnmupdateJdk.ovpl <path>`
 - *On Linux:* `/opt/OV/bin/nnmupdateJdk.ovpl <path>`

In this instance, `<path>` is the path to the home directory of the non-default JDK.

The command also updates the `NMS_JAVA_HOME` environment variable with the reference of the new JDK home directory.

Change JDK on NNMi in an application failover cluster

1. Note the `com.hp.ov.nms.cluster.name` property value in the `nms-cluster.properties` file. You will need this value later. This file is in the following location:
 - Windows: `%nnmDataDir%\shared\nnm\conf\props\nms-cluster.properties`
 - Linux: `$nnmDataDir/shared/nnm/conf/props/nms-cluster.properties`
2. Run `nnmcluster` on one of the nodes.
3. Enter `dbsync` on the NNMi management server used in the previous step to synchronize the two databases.

Note: The `dbsync` option works on an NNMi management server using the embedded database. Do not use the `dbsync` option on an NNMi management server configured to use an Oracle database.

4. Wait until the active NNMI management server reverts to ACTIVE_NNM_RUNNING and the standby NNMI management server reverts to STANDBY_READY. before continuing.
5. Exit or quit from the `nnmcluster` command.
6. Stop the cluster on the standby NNMI management server by running the following command on the standby NNMI management server:
`nnmcluster -shutdown`
7. Make sure the following processes and services terminate before continuing:
 - postgres
 - ovjboss
8. Make sure the `nnmcluster` process terminates before continuing. If the `nnmcluster` process will not terminate, manually kill the `nnmcluster` process only as a last resort.
9. 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
 - a. Comment out the cluster name by placing a # at the front of the line, then save your changes:
`#com.hp.ov.nms.cluster.name = NNMIcluster`
10. Run the following command:
 - *On Windows:* %nnminstalldir%\bin\nnmupdateJdk.ovpl <path>
 - *On Linux:* /opt/OV/bin/nnmupdateJdk.ovpl <path>

In this instance, <path> is the path to the home directory of the non-default JDK.
The command also updates the NMS_JAVA_HOME environment variable with the reference of the new JDK home directory.
11. Stop the active NNMI management server and immediately bring the standby NNMI management server online to monitor your network.
12. Shut down the cluster on the active NNMI management server by running the following command on the active NNMI management server:
`nnmcluster -halt`
13. Make sure the `nnmcluster` process terminates. If it does not terminate within a few minutes, manually kill the `nnmcluster` process.
14. On the standby NNMI management server, uncomment the cluster name from the `nms-cluster.properties` file.

Note: During patch installation the `com.hp.ov.nms.cluster.name` property value is replaced with the NNMI default value. After you uncomment the line that contains the `com.hp.ov.nms.cluster.name` parameter, you also need to replace the `com.hp.ov.nms.cluster.name` property value with the value that was configured before the patch was installed.

- 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 in the `nms-cluster.properties` file on the active NNMI management server.
 - c. Replace the default value of the `com.hp.ov.nms.cluster.name` property with the name that was configured in `nms-cluster.properties` before the command was run.
 - d. Save your changes.
15. Start the cluster on the standby NNMI management server by running the following command on the standby NNMI management server:
nnmcluster -daemon
16. Run the command on the active NNMI management server.
 - *On Windows:* `%nnminstallDir%\bin\nnmupdateJdk.ovpl <path>`
 - *On Linux:* `/opt/OV/bin/nnmupdateJdk.ovpl <path>`
17. At this point, the previous active NNMI management server is offline. Bring it back into the cluster (as the standby NNMI management server) by performing the following:
 - a. Uncomment the `com.hp.ov.nms.cluster.name` parameter in the `nms-cluster.properties` file on the active NNMI management server.
 - b. Replace the default value of the `com.hp.ov.nms.cluster.name` property with the name that was configured in `nms-cluster.properties` before the patch was installed.
 - c. Start the active NNMI management server using the following command:
nnmcluster -daemon
18. To monitor the progress, run the following command on both the active and standby NNMI management servers:
nnmcluster
Wait until the previous active NNMI management server finishes retrieving the database from the previous standby NNMI management server.
19. After the previous active NNMI management server displays `STANDBY_READY`, run the following command on the previous active NNMI management server:
nnmcluster -acquire

Change JDK on NNMI in an HA cluster

To change JDK for NNMI, work in High Availability (HA) maintenance mode. Follow these steps:

1. Determine which node in the HA cluster is active:
 - *Windows:*
`%NnmInstallDir%\misc\nnm\ha\nnmhaclusterinfo.ovpl -group <resource_group> -activeNode`
 - *Linux:*
`$NnmInstallDir/misc/nnm/ha/nnmhaclusterinfo.ovpl -group <resource_group> -activeNode`
2. On each passive node, put the NNMI HA resource group into maintenance mode as described in ["Putting an HA Resource Group into Maintenance Mode" on page 177](#).
Include the `NORESTART` keyword.
3. On each passive node, change the JDK by following the steps in [Change JDK on a standalone NNMI](#)

management server.

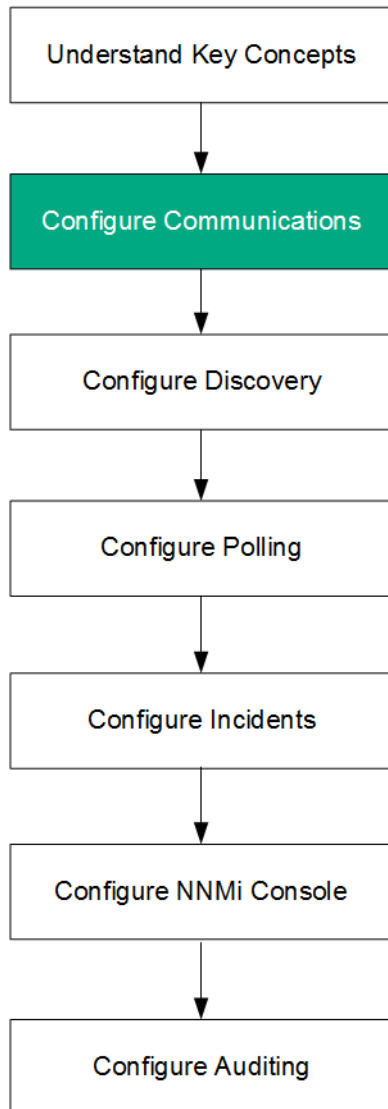
Caution: Never run the `ovstart` or `ovstop` commands on a secondary (backup) cluster node.

4. On all passive nodes, take the NNMi HA resource group out of maintenance mode as described in ["Removing an HA Resource Group from Maintenance Mode" on page 177](#).
5. Fail over to a passive node.
6. Go to the node that was previously active (in [step 1](#)), and then follow these steps:
 - a. Put the NNMi HA resource group of the node into maintenance mode as described in ["Putting an HA Resource Group into Maintenance Mode" on page 177](#).
Include the `NORESTART` keyword.
 - b. On the node, change the JDK by following the steps in [Change JDK on a standalone NNMi management server](#).

Caution: Never run the `ovstart` or `ovstop` commands on a secondary (backup) cluster node.

- c. On the node, take the NNMi HA resource group out of maintenance mode as described in ["Removing an HA Resource Group from Maintenance Mode" on page 177](#).

NNMi Communications



HPE Network Node Manager i Software (NNMi) uses Simple Network Management Protocol (SNMP) and Internet Control Message Protocol (ICMP ping) to discover devices and to monitor device status and health.

Note: If Web Agents are configured (in addition to SNMP Agents), NNMi can use additional protocols. For example, **SOAP**¹ protocol for **VMware**² environments.

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.

¹Simple Object Access Protocol

²VMware ESX and VMware ESXi software uses SOAP protocol to implement bare-metal hypervisors.

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.

Caution: If your devices use either SNMP v1 or SNMP v2C, note the following:

- SNMP v1 and SNMP v2C send their information packets in clear text.
- To secure your environment, use SNMP v3 or add protections, such as firewall controls, for the flow of SNMP traps and the collection of information from your devices.

This chapter contains the following topics:

- "Concepts for Communications" below
- "Plan Communications" on page 41
- "Configure Communications" on page 44
- "Evaluate Communications" on page 53
- "Tune Communications" on page 54

Concepts for Communications

NNMI uses SNMP and ICMP primarily in a request-response manner. Responses to ICMP ping requests verify address responsiveness. Responses to other management protocols, such as 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**¹ protocol for **VMware**² environments.

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 any settings are not yet defined, 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

¹Simple Object Access Protocol

²VMware ESX and VMware ESXi software uses SOAP protocol to implement bare-metal hypervisors.

values specified for that region to fill in the blanks left from the applicable specific node setting (if any). The settings for additional regions are not considered.

3. If any settings are still not yet defined, 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.

Network Latency and Timeouts

Normal network latency influences the amount of time the NNMi management server must wait to get answers to ICMP 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 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 between each retry. For example, when NNMi is configured to use timeout of 5 seconds and three retries, NNMi waits 5 seconds for a response to the first request, 10 seconds for a response to the second request, and 15 seconds for a response to the third request before giving up until the next polling cycle.

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 hash-based message authentication code (HMAC) using your choice of either the message-digest algorithm 5 (MD5) or the secure hash algorithm (SHA).
- Privacy protocol uses no encryption or the data encryption standard - cipher block chaining (DES-CBC) symmetric encryption protocol.

Note: DES-CBC is considered a weak cipher. Therefore, if you are using DES-CBC, HPE recommends that you choose a stronger cipher. To change your cipher selection:

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. In the **Privacy Protocol** list, select a strong cipher.

Note: Avoid using DES-CBC when configuring SNMPv3 communication on the nodes that NNMi manages.

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

Also see ["SNMP Access Control in High Availability \(HA\) Environments"](#) on page 1

SNMP Access Control in High Availability (HA) Environments

When NNMi is configured in a High Availability (HA) environment, the SNMP source address is set to a physical cluster node address. To set the SNMP source address to the NNM_INTERFACE (which is set to the virtual IP address), you must edit the `ov.conf` file and set the value for `IGNORE_NNM_IF_FOR_SNMP` to OFF. (By default, this setting is set to ON.)

To set the SNMP source address to the NNM_INTERFACE in HA environments:

1. Edit the following file on both nodes in the cluster:
 - Windows:* %NnmDataDir%\shared\nnm\conf\ov.conf
 - Linux:* \$NnmDataDir/shared/nnm/conf/ov.conf
2. Set the value for `IGNORE_NNM_IF_FOR_SNMP` to OFF. (By default, this setting is set to ON.)


```
IGNORE_NNM_IF_FOR_SNMP=OFF
```
3. Stop and restart the NNMi management server:

Note: Put the nodes in maintenance mode before running the `ovstop` and `ovstart` commands

- a. Run the `ovstop` command on the NNMi management server.
- b. Run the `ovstart` command on the NNMi management server.

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.

Note: 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 SNMPv2c, NNMi discards incoming SNMPv3 traps. If these traps must be received, follow the procedure in *Configuring NNMi to Authenticate SNMPv3 Traps for Nodes Not Being Monitored*.

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**—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.

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 information about how NNMi determines the management address, see *Node Form* in the NNMi help.

Note: 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.

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

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

SNMPv3 Traps and Informs

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

When the device sends a trap to NNMi, the device may not have the NNMi information, and because a trap is a single-packet transaction, it has no way to get the necessary information. Therefore, it uses its own Engine ID, boot count and engine time in the trap, along 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 per 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 the 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.

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. (However, if the device responds to SNMP, State Poller does not use ICMP.)
- Operators cannot use **Actions > 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.

Communication Configuration and the `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 `ovjboss` is not running, the `nnmsnmp*.ovpl` commands behave as follows:

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

Plan Communications

Make decisions in the following areas:

- ["Default Communication Settings" below](#)
- ["Communication Configuration Regions" below](#)
- ["Specific Node Configurations" on the next page](#)
- ["Retry and Timeout Values" on page 43](#)
- ["Active Protocols" on page 43](#)
- ["Multiple Community Strings or Authentication Profiles" on page 43](#)

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.

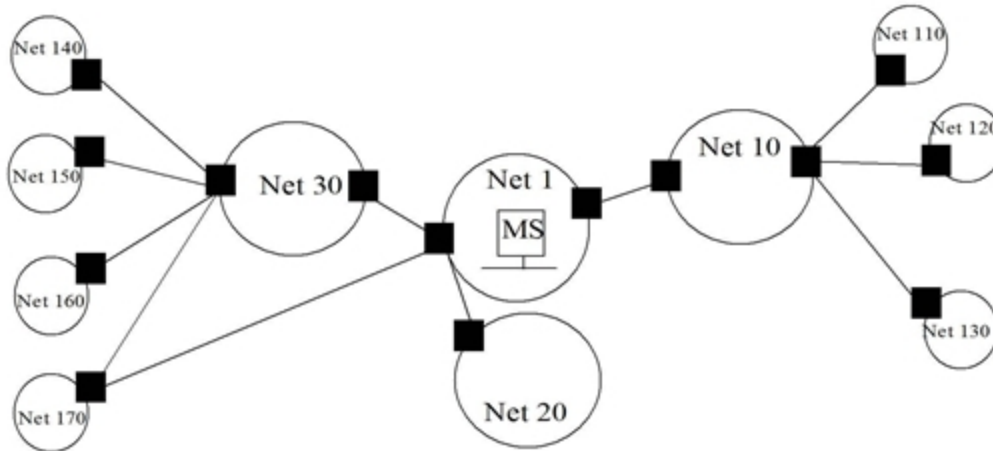
- Are there commonly-used community strings that NNMi should try?
- What default timeout and retry values are reasonable in your network?

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:

Network Example for Communication Regions



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
- 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 or 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 value 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).

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

Note: You can enable or disable SNMP communication for a specific device. See *Specific Node Settings Form* in the NNMi help.

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 current lag time for each hop, do the following:

- *Windows*: Run a `tracert` to a device in each network area.
- *Linux*: Run a `traceroute` to a device in each network area.

Active Protocols

You have two opportunities to control the type of traffic NNMi generates when communicating with devices in your network: 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 or monitoring settings disable a protocol for a device, NNMi does not generate that type of traffic to the device.

Note: Disabling SNMP communication significantly compromises the NNMi status and health monitoring of your network.

Note whether each region or specific device should 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).

Also see "[Device Support Using the Network Configuration Protocol \(NETCONF\)](#)" on page 46.

Multiple Community Strings or 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 your regions (and the associated community strings and authentication protocols to try) as tightly as possible.

If your environment uses SNMPv1 or v2c *and* SNMPv3, determine the minimum acceptable security level for each region.

SNMPv1 and SNMPv2 Community Strings

For regions where SNMPv1 or v2c access is acceptable, gather the community strings in use within the region and any unique community strings required by specific devices.

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.

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

You can specify one (or no) authentication protocol and one (or no) privacy protocol for each specific node or region setting.

Note: Use of the TripleDES, AES-192, or AES-256 privacy protocols requires the Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy Files library, which is installed automatically as part of the NNMi installation process. If you accidentally delete the library, you can restore it by following the procedure in ["NNMi Configuration Issues" on page 467](#).

Configure Communications

After reading the information in this section, see *Configuring Communication Protocol* in the 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.

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.

Note: Double-check the ordering numbers for the defined regions. If a node qualifies for membership in

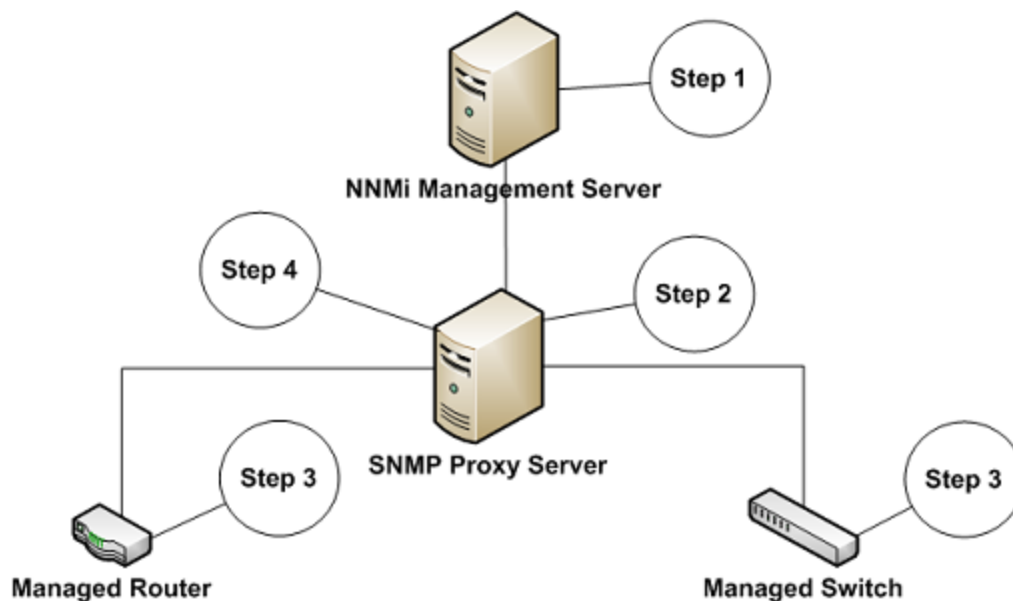
multiple regions, NNMi applies the settings from the region with the lowest ordering number to that node.

Configuring SNMP Proxy Settings

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.

Tip: For the alternate method of configuring SNMP proxy settings from the command line, see the `nmcommunication.ovpl` reference page.

Using Proxy Servers



1. The NNMi management server sends an SNMP request to an SNMP proxy address and SNMP proxy port to obtain information from the managed router and the managed switch.
2. The SNMP proxy server determines where to send the SNMP request, then sends SNMP requests to the managed router and switch to obtain the information requested by the NNMi management server.
3. The managed switch and router respond to the SNMP proxy server (using the SNMP Proxy Address and SNMP Proxy Port) with 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.

Device Support Using the Network Configuration Protocol (NETCONF)

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

Currently, NNMi uses NETCONF to support Juniper Networks QFabric systems only. See the HPE Network Node Manager i Software Device Support Matrix for any updates.

The following sections provide a brief introduction to NETCONF and information about the configuration required for both the managed device and NNMi:

["What is Network Configuration Protocol \(NETCONF\)?" below](#)

["Network Configuration Protocol \(NETCONF\) Operations" on the next page](#)

["Enabling and Configuring Network Configuration Protocol \(NETCONF\) in a Managed Device" on the next page](#)

["Configuring Network Configuration Protocol \(NETCONF\) Device Credentials in NNMi" on the next page](#)

What is Network Configuration Protocol (NETCONF)?

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

NNMi uses NETCONF to collect information about the device during discovery or rediscovery (in other words, read-only information). NNMi does not use NETCONF to modify device configurations or to monitor status or 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 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 device.

NETCONF is a relatively new management protocol; therefore, it is not as widely available across device vendors as compared to SNMP.

If a vendor implements NETCONF in a device that NNMi is managing, note the following:

- NETCONF commands are generally more vendor specific and are not as well publicized as the many standard and vendor-specific MIBs in SNMP. Consequently, the ability for NNMi to make use of 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. See ["Enabling and Configuring Network Configuration Protocol \(NETCONF\) in a Managed Device"](#) below and ["Configuring Network Configuration Protocol \(NETCONF\) Device Credentials in NNMi"](#) below for more information.

Network Configuration Protocol (NETCONF) Operations

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

- A NETCONF client (management application, such as NNMi) establishes an SSH connection with the NETCONF server (subsystem) on the managed device. Valid SSH user name and password credentials must be specified by the client and authenticated by the device.
- The client application and device exchange capabilities in the form of <hello> messages.
- 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.
- The device responds with results of the operations in the form of RPC reply messages.
- When the client application has finished sending requests and processing the responses, it sends a <close-session> RPC message to the device.
- The device acknowledges with an <ok> RPC reply message.
- Finally, both sides terminate the SSH connection.

Enabling and Configuring Network Configuration Protocol (NETCONF) in a Managed Device

You might need to explicitly enable and configure NETCONF in the managed device before NNMi is able to communicate with that device. See your vendor's device configuration documentation for specific instructions. For example, for Juniper Networks QFabric Systems, see "Establishing a NETCONF Session" in Juniper Networks' NETCONF XML Management Protocol Guide.

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 the SSH user name and password credentials on the device for NETCONF communication access. NNMi requires only read-only access.

See the HPE Network Node Manager i Software Device Support Matrix ("Known Limitations" section) for the current list of supported devices using NETCONF in NNMi, plus any additional vendor-specific prerequisites and references.

Configuring Network Configuration Protocol (NETCONF) Device Credentials in NNMi

You must configure NETCONF SSH credentials in NNMi to match those configured in the managed device before NNMi is able to communicate with that device using NETCONF.

Note: 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.

Use the NNMi console to configure NETCONF device credentials settings in the **Communication Configuration, Device Credentials** tab of the relevant Node-specific Settings, Region Settings, or Default Settings for the device.

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

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

See the NNMi Help for Administrators for detailed instructions about how to edit the NNMi **Communication Configuration** forms.

Configuring Communication for Virtual Environments

This section describes configuration information to enable NNMi communicate with supported virtual environments.

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 hosted on the hypervisors:

Pre-Requisites for Monitoring hypervisor and its VMs

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 <i>Help for Administrators > Configuring Communication Protocol</i> , see instructions for SNMP settings for Default, Regions, or Specific Nodes.

Pre-Requisites for Monitoring hypervisor and its VMs, continued

What you want to discover?	Prerequisite(s)	For more information
	<p>NNMi must be configured to communicate with the hypervisor using HTTPS.</p> <p>Note: <i>VMware only.</i> 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. For example steps to be followed on ESX5.1 and ESX5.5 servers, see "Replacing the VMware Default Certificate" below</p>	<p>To configure using CLI, see the <code>nnmcommunication.ovpl</code> reference page, or the Linux manpage, for more information.</p> <p>To configure using CLI, see "Configuring NNMi to Communicate with Hypervisors Using HTTPS" on the next page.</p> <p>To configure using NNMi user interface, see <i>Help for Administrators > Configuring Communication Protocol</i>, instructions for Trusted Certificate Settings for Default, Regions, or Specific Nodes.</p>
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.	<p>To configure using NNMi user interface, see <i>Help for Administrators > Configuring Communication Protocol</i>, instructions for Credential Settings for Default, Regions, or Specific Nodes.</p> <p>To configure using CLI, see <code>nnmcommunication.ovpl</code> reference page or the Linux manpage.</p>

Replacing the VMware Default Certificate

Note: 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.

To replace the VMware default certificate with a certificate that is generated using the hostname of the ESXi server, follow these example steps on the ESXi server:

Note: This example describes the steps to be followed on ESX5.1 and ESX5.5 servers. For the latest information, see the VMware documentation that describes how to replace the VMware default

certificate.

1. Make sure the `/etc/hosts` file has the following format for resolving the host:


```
#/etc/hosts
127.0.0.1 localhost.localdomain localhost
::1 localhost.localdomain localhost
10.78.xx.xxx hostname.domain.com hostname
```
2. Make sure SSH is enabled on the ESXi server.
3. Log in to the ESXi Shell as a user with administrator privileges.
4. Navigate to following directory:


```
/etc/vmware/ssl
```
5. Back up any existing certificates by renaming them using the following commands:


```
mv rui.crt orig.rui.crt
mv rui.key orig.rui.key
```
6. To generate new certificates, run the following command:


```
/sbin/generate-certificates
```
7. Restart the host.
8. Confirm the host successfully generated new certificates:
 - a. Use the following command to list the certificates:


```
ls -la
```
 - b. Compare the time stamps of the new certificate files with `orig.rui.crt` and `orig.rui.key`. If the original files are available.

Configuring NNMi to Communicate with Hypervisors Using HTTPS

Note: If you need to use HTTP to communicate with hypervisors, also see ["Enable HTTP to Communicate with Hypervisors" on page 52](#).

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).

Note: 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.

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

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.

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. For example steps to be followed on ESX5.1 and ESX5.5 servers, see ["Replacing the VMware Default Certificate " on page 49](#)

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.	<code>nnmcommunication.ovpl addCertificate -default -cert <fully qualified path to the certificate file></code>
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.	<code>nnmcommunication.ovpl addCertificate -region <region name or UUID> -cert <fully qualified path to the certificate file></code>
Node	To upload an SSL certificate (CA or Self-Signed server 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.	<code>nnmcommunication.ovpl addCertificate -nodeSetting <node name or UUID> -cert <fully qualified path to the certificate file></code>

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.

Tip:

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

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 the
VMware vSphere API.
```

```
#HPE 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 **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.

Note: Follow the steps described in ["Configuring NNMi to Communicate with Hypervisors Using HTTPS" on page 50](#).

Evaluate Communications

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

Consider the following:

- ["Are All Nodes Configured for SNMP?" below](#)
- ["Is SNMP Access Currently Available for a Device?" below](#)
- ["Is the Management IP Address for SNMP Devices Correct?" below](#)
- ["Is NNMi Using the Correct Communications Settings?" below](#)
- ["Do the State Poller Settings Agree with the Communication Settings?" on the next page](#)

Are All Nodes Configured for SNMP?

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

Is SNMP Access Currently Available for a Device?

1. Select the node in an inventory view.
2. Select **Actions > Status Poll** or **Actions > Configuration Poll**.
If the results show any SNMP values, communication is operational.

You can also test communication from the command line with the `nmmsnmpwalk.ovpl` command. For more information, see the `nmmsnmpwalk.ovpl` reference page, or the Linux manpage.

Is the Management IP Address for SNMP Devices Correct?

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

1. Select the node in an inventory view.
2. Select **Actions > Communication Settings**.
3. On the **Communication Configuration** form, verify that the management address of the SNMP agent listed in the Active SNMP Agent Settings list is correct.

Is NNMi Using the Correct Communications Settings?

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

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

1. Select the node in an inventory view.
2. Select **Actions > Communication Settings**.
3. On the **Communication Configuration** form, 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 Administrators.

Do the State Poller Settings Agree with 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. Select **Actions > Monitoring Settings**.

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

Tune Communications

Reduce authentication failures

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

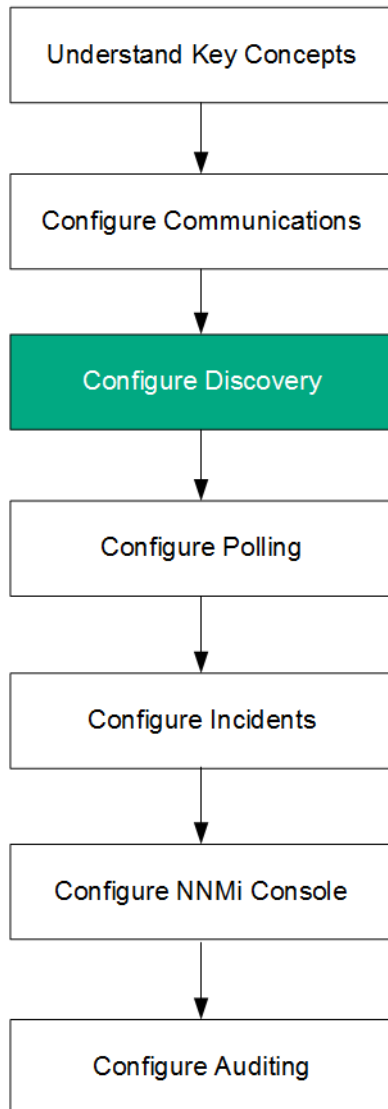
Tune timeouts and retries

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

Reduce 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.

NNMi Discovery



One of the most important network management tasks is keeping your view of the network topology current. HPE Network Node Manager i Software (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 the NNMi help.

This chapter contains the following topics:

- ["Concepts for Discovery" on the next page](#)
- ["Plan Discovery" on page 57](#)
- ["Configure Discovery" on page 65](#)

- ["Evaluate Discovery" on page 67](#)
- ["Tune Discovery" on page 71](#)

Concepts for Discovery

The NNMi default behavior of 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. Generally, you should avoid managing end nodes (for example, personal computers or printers) unless the end node is identified as a critical resource. For example, database and application servers might be considered critical resources.

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

Note: 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**¹: Each device with a Device Profile of vwareVM is equivalent to 1/10th node.
 - All other devices are equivalent to one discovered node.

For details about license limits, see "Track Your NNMi Licenses" in the NNMi Help for Administrators.

- 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.

Note: For information about configuring Discovery to discover a large number of nodes, see the 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 the devices beyond the edge of your responsibility. (For information about the State Poller, see ["NNMi State Polling" on page 73.](#))

NNMi provides two primary discovery configuration models:

¹VMware ESX and VMware ESXi software uses SOAP protocol to implement bare-metal hypervisors.

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

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

Note: NNMi uses tenancy to support networks with overlapping address domains that may exist within static Network Address Translation (NAT), dynamic Network Address Translation (NAT), or Port Address Translation (PAT) areas of your network management domain. If you have such networks, put the overlapping address domains into different tenants (this is done using seeded discovery). See the NNMi help for more information.

Note: If you are using NNMi to manage VMware Hypervisor-Based Virtual Networks, see the "Tenants within Virtual Environments" help topic in the Help for Administrators.

Tip: If you plan to configure multi-tenancy, configure tenants before initiating network discovery.

NNMi Derives Attributes through Device Profiles

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

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

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 thousands 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.

Plan Discovery

Make decisions in the following areas:

- ["Select Your Primary Discovery Approach" on the next page](#)
- ["Auto-Discovery Rules" on page 59](#)
- ["Node Name Resolution" on page 61](#)
- ["Subnet Connection Rules" on page 62](#)

- ["Discovery Seeds" on page 62](#)
- ["Rediscovery Interval" on page 63](#)
- ["Do Not Discover Objects" on page 63](#)
- ["Discover Interface Ranges" on page 64](#)
- ["Monitor Virtual IP Addresses with NNMi" on page 64](#)
- ["Use Discovery Hints from SNMP Traps" on page 65](#)

Select Your Primary Discovery Approach

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

List-Based Discovery

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

Note: NNMi uses tenancy to support networks with overlapping address domains that may exist within static Network Address Translation (NAT), dynamic Network Address Translation (NAT), or Port Address Translation (PAT) areas of your network management domain. If you have such networks, put the overlapping address domains into different tenants (this is done using seeded discovery). See the NNMi help for more information.

Note: If you are using NNMi to manage VMware Hypervisor-Based Virtual Networks, see the "Tenants within Virtual Environments" help topic in the Help for Administrators.

Tip: If you plan to configure multi-tenancy, list-based discovery is the recommended discovery approach.

Benefits of using only list-based discovery 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 only list-based discovery include:

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

Rule-Based Discovery

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

Benefits of using rule-based discovery include:

- Good for large networks. NNMi can discover a large number of devices based on minimal configuration input.
- Good for networks that change frequently. New devices that are added to the network are discovered without administrator intervention (assuming that each device is covered by an auto-discovery rule).
- Ensures that any new device added to your network is discovered to comply with service level agreements for managing new devices in a timely manner or security guidelines to flag unauthorized new devices.

Disadvantages of using rule-based discovery include:

- It is easier to run into license limitations.
- Depending on the structure of your network, tuning auto-discovery rules can be complex.
- If auto-discovery rules are very broad and NNMi discovers many more devices than you want to manage, you might want to delete the unneeded devices from NNMi topology. 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.

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

Auto-Discovery Rule Ordering

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

- IP address ranges

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

Exclude Devices from Discovery

- To prevent discovery of certain object types, create an auto-discovery rule with a low ordering number that ignores the system object IDs that you do not want discovered. Do not include an IP address range in this rule. By giving this auto-discovery rule a low ordering number, the discovery process quickly passes by the objects that match this rule.

- The **Ignored by Rule** setting for an IP address range or a system object ID range affects that auto-discovery rule only. The devices included in an ignored range are available to be included in another auto-discovery rule.

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 an router redundancy group (RRG), as they are when using HSRP, the routers configured in the RRG share a protected IP address (one active and one standby). NNMi does not support the discovery and management of multiple RRGs configured with the same protected IP address. Each RRG must have a unique protected IP address.

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.

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

Tip: Enable ping sweep for small discovery ranges only.

Discovery Seeds for Auto-Discovery Rules

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

- Enter seeds on the **Discovery Seed** form by clicking **Seeds** under **Discovery** in the **Configuration** workspace.
- Use the `nnmloadseeds.ovp1` 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.

Best Practices for Auto-Discovery Rules

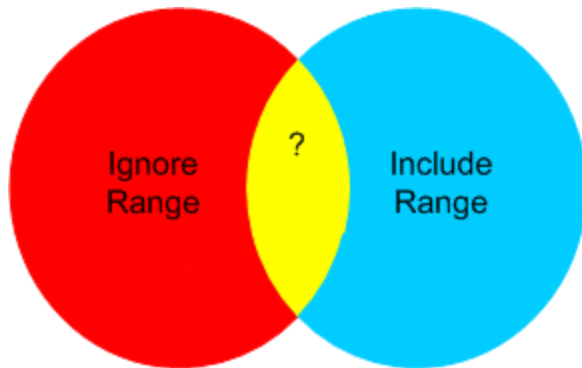
- Because NNMi automatically manages all discovered devices, use IP address ranges that closely match the areas of the network that you want to manage.
 - You can use multiple IP address ranges within an auto-discovery rule to restrict discovery.
 - You can add a large IP address range to an auto-discovery rule and then exclude some IP addresses from discovery within that rule.

- The system object ID range specification is a prefix, not an absolute value. For example, the range 1.3.6.1.4.1.11 is the same as 1.3.6.1.4.1.11.*.

Discovery Rule Overlap

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

Overlapping Discovery Ranges



Limit Device Type Discovery

To discover all HPE devices in your network that are not printers, create one auto-discovery rule with a range to include the HPE enterprise system object ID (1.3.6.1.4.1.11). In this auto-discovery rule, create a second range to ignore the system object IDs of HP printers (1.3.6.1.4.1.11.2.3.9). Leave the IP address range unset.

Node Name Resolution

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

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

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

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

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

Tip: 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.

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

Tip: 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.

Subnet Connection Rules

List-based discovery only

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

Rule-based discovery only

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

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

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

Discovery Seeds

List the devices to use as discovery seeds.

Tip: 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.

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

List-based discovery only

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

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

Rule-based discovery only

Discovery seeds are optional for rule-based discovery:

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

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

Rediscovery Interval

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

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

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

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

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

Do Not Discover Objects

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

- On the **Communication Configuration** form, you can turn off ICMP communication, SNMP communication, or both at different levels: globally, for communication regions, or for specific hostnames or IP addresses.
- On the **Discovery Configuration** form, you can set up an auto-discovery rule that instructs NNMi to never gather hints from certain IP addresses or SNMP system object IDs. Nodes matching the criteria still appear on the map and in the database, but spiral discovery does not extend to the neighboring devices beyond those IP addresses or object types.
- On the **Discovery Configuration** form, you can set up an auto-discovery rule that instructs NNMi to exclude specific IP address ranges, IP addresses, or both from the database. Spiral discovery does not display those addresses on any node's list of addresses or use those addresses when establishing connections between devices, so NNMi never monitors the health of those addresses.
- On the **Excluded IP Addresses** tab of the **Discovery Configuration** form, you can exclude a range of IP addresses from being discovered by configuring an excluded IP addresses filter.

If all of a node's IP addresses are entered into the Excluded IP Addresses list after that node was already 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.

Note: If you exclude an IP address range, any duplicates of addresses in static Network Address Translation (NAT), dynamic Network Address Translation (NAT), or Port Address Translation (PAT) areas of your network management domain are also excluded.

NNMi uses tenancy to support networks with overlapping address domains. If you have such networks, put the overlapping address domains into different tenants (this is done using seeded discovery). See the NNMi help for more information.

- On the **Excluded Interfaces** tab of the **Discovery Configuration** form, you can exclude a certain type of interface from the discovery process by selecting an Interface Group. See the NNMi help for more information.

Discover Interface Ranges

NNMi enables you to specify a range of interfaces to be discovered by defining a filter. This is particularly helpful when you have large nodes where you only want to discover a subset of the interfaces. When you specify a range of interfaces to be discovered, NNMi does not ask for information about interfaces outside that range; whereas, using the excluded interface option filters interfaces after retrieving the information from the device. Therefore, range-based discovery can improve discovery performance for large devices, especially when you do not want to manage all the interfaces on such devices.

The included interface ranges filter, defined on the **Included Interface Ranges** tab of the **Discovery Configuration** form, uses the System Object ID prefix and the ifIndex values to define the interface range. See the NNMi help for more information.

Monitor Virtual IP Addresses with NNMi

NNMi discovers and monitors devices such as clustered servers that share a virtual IP address. After a cluster fails over to a new active node, NNMi associates the virtual IP address with the new active node. This association is not immediate, as some time might pass between failover and NNMi discovering the change.

You can take several actions to configure NNMi for your specific situation:

If you want NNMi to monitor a virtual IP address, *use only one of the following options*:

- Option 1: For this option, NNMi manages N+1 non-SNMP devices, where N represents the number of members in the cluster discovered with a non-virtual IP address. NNMi discovers the additional (+1) non-SNMP node, and it is configured with the virtual IP address.
Do nothing to stop NNMi from discovering a virtual IP address. Using this approach, NNMi discovers the virtual IP address and the physical IP addresses associated with the Network Interface (NIC) cards on devices configured to use this virtual IP address. NNMi discovers and monitors each device as a separate non-SNMP node.
- Option 2: Configure NNMi to use a device's physical IP address as the Preferred Management Address of a clustered server. For instructions on how to do this, see the *Specific Node Settings Form (Communication Settings)* topic in the NNMi help.

Note: NNMi might not immediately recognize the transfer of a virtual IP address from one active node to a new active node. NNMi might show the status of a virtual IP address using a node other than the current active node in the cluster.

If you do not want NNMi to monitor a virtual IP address, do the following using the NNMi console:

1. Click **Discovery Configuration** in the **Configuration** workspace.
2. Click the **Excluded IP Addresses** tab.
3. Add the virtual IP address or range of addresses to the list of addresses to be excluded from discovery.
4. Save your changes.

Use Discovery Hints from SNMP Traps

NNMi processes the source IP address of all incoming SNMP traps as hints to NNMi auto-discovery rules. See the NNMi *Help for Administrators* for more information about SNMP Trap Incidents.

Configure Discovery

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

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

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

Tip: It is a good idea to save a copy of the existing configuration before you make any major configuration changes.

Tips for Configuring Auto-Discovery Rules

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

Tips for Configuring Seeds

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 `nmmLoadseeds.ovp1` command to import the node list into NNMi.
- In the seed file, specify IP addresses as a way of influencing the IP address that NNMi chooses as the management address. (If you use hostnames, DNS provides the IP address for each node.)
- Good formats for the entries in the seed file are shown here:

```
IP_address1 # node name
```

```
IP_address2, <tenant_UUID_or_tenant_name> # node name
```

These formats are easy for both NNMi and human readers.

- For maintenance purposes, it is better to use only one seed file. Add nodes as needed and then rerun the `nnmloadseeds.ovp1` command. NNMi discovers the new nodes but does not re-evaluate the existing nodes.

Note: If the seed file cannot be loaded, try making the file readable by `nmsproc` (644 permissions).

- Removing a node from the seed file does not remove it from the NNMi topology. Delete the node directly in the NNMi console.
- Deleting a node from a map or inventory view does not delete the seed.
- If you want NNMi to rediscover a node, delete that node from a map or inventory view *and* from the **Seeds** form in the **Discovery** area of the **Configuration** workspace in the NNMi console, and the re-enter the node in the NNMi console, or run the `nnmloadseeds.ovp1` command.

Rule-based discovery only

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

Discovering Link Aggregation

Note: Link Aggregation requires an NNMi Advanced or NNMi Premium license.

Link Aggregation (LAG) protocols enable network administrators to configure a set of interfaces on a switch as one Aggregator Interface. This 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.

Search for **Link Aggregation** in the NNMi Help for more information.

Discovering Server-to-Switch Link Aggregations (S2SLA)

Note: Link Aggregation requires an NNMi Advanced or NNMi Premium 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 automatically negotiated 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, the 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-to-server connections. To ensure that NNMi can properly discover S2SLA information for a node, complete the following tasks:

- By default, Linux does not install its SNMP agent package, Net-SNMP. 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.

Tip: All interfaces in the aggregation use the same MAC address. A walk 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 show 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
```

Evaluate Discovery

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

Follow the Progress of Initial Discovery

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

- 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** page. Refresh this page until all seeds show the `Node created` results, which indicates that the device has been added to the topology database. This result does *not* indicate that NNMi has gathered all information from the device and processed its connectivity.
- Open the **Node** form for representative nodes. When the **Discovery State** field (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 are present from different areas of your network.
- 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.

Were All Seeds Discovered?

1. From the **Configuration** workspace, under **Discovery**, click **Seeds**.
2. On the **Seeds** page, 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 or IP address**—For these types of failures, verify network connectivity to the node and check for accurate DNS name resolution. To work around DNS issues, use the IP address to seed the node or include the hostname in a `hostnolookup.conf` file. For problems due to IP addresses that should not be resolved to hostnames, include the IP addresses in a `ipnolookup.conf` file. See the `hostnolookup.conf` and `ipnolookup.conf` reference pages, or the Linux manpages, for more information.
 - **License node count exceeded**—This scenario occurs when the number of devices already discovered reached your license limit. You can either delete some discovered nodes or purchase additional node pack licenses.

When tracking license information, note the following:

- **Consumption:** NNMi discovers and manages nodes up to the NNMi licensed capacity limit (rounded up):
 - **VMware¹**: Each device with a Device Profile of `vmwareVM` is equivalent to 1/10th node.
 - All other devices are equivalent to one discovered node.

For details about license limits, see “Track Your NNMi Licenses” in the NNMi Help for Administrators.

- **Node discovered but no SNMP response**—SNMP communication problems can occur for seeded devices as well as devices that are discovered through auto-discovery. For more information, see ["Evaluate Communications" on page 53](#).

Do All Nodes Have a Valid Device Profile?

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

Were All Nodes Discovered Properly?

To avoid discovery problems, NNMi should only manage nodes using a unique IP address that does not appear on 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.

Note: NNMi does not properly manage a router if it tries to use a protected IP address as the management address.

¹VMware ESX and VMware ESXi software uses SOAP protocol to implement bare-metal hypervisors.

Examine the data in the **Nodes** inventory view. If any nodes do not have a management address, check the communication settings for those nodes as described in ["Are All Nodes Configured for SNMP?" on page 53](#).

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

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

Auto-Discovery Rules

List-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 if a node should be created. If no rules are matched, NNMi discovery discards the hint. The ordering number for auto-discovery rules determines the order in which the auto-discovery rule configuration settings are applied.

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

- **Discover Included 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. Enabling these settings without considering your environment can result in NNMi discovering more nodes than intended.

IP Address Ranges

The IP address of a discovery hint must match an **Include in Rule** entry in the IP address range list. If there are no included IP address ranges in an auto-discovery rule, then all address hints are considered a match. (For this case, see ["Tips for Configuring Auto-Discovery Rules" on page 65](#).) Additionally, the hint must *not* match any entry marked **Ignored by Rule**. If all checks successfully match, this rule's configuration is used for handling the hint.

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

System Object ID Ranges

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

- Use the system object ID ranges to either expand auto-discovery to include more than the default routers and switches, or to exclude specific routers and switches.

- Each node must match both the IP address range and the system object ID range specified before it is discovered and added to the topology database.

Are All Connections and VLANs Correct?

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

Evaluate Layer 2 Connectivity

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

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

NNMi Discovery and Duplicate MAC Addresses

Discovery takes MAC Addresses into account for the following benefits:

- Improves support for DHCP or other nodes that change IP addresses.
- Improves node identity for nodes configured with duplicate IP addresses.
- Improves support for devices that do not report hosted IP addresses.

During discovery, NNMi reads the Forwarding Database (FDB) tables from Ethernet switches within a network to help NNMi determine communication paths between network devices. NNMi searches these FDB tables for information about discovered nodes. When an NNMi management server finds FDB references to duplicate Media Access Control (MAC) addresses, it does the following:

- If two or more discovered nodes contain an interface associated with the same Media Access Control (MAC) address within the same Tenant or with one of those nodes in Default Tenant and one in any other Tenant, NNMi disregards the communication paths reported for those duplicate MAC addresses in the FDB. This might result in missing connections on NNMi maps in network areas that include those duplicate MAC addresses.

NNMi Advanced or NNMi Premium - Global Network Management feature: If two NNMi management servers discover nodes that contain an interface associated with the same Media Access Control (MAC) address, the Global NNMi management server's maps could be missing connections that are visible on the Regional NNMi management server's maps.

- If a single node contains multiple interfaces that have the same MAC address, NNMi gathers all communication path information for those interfaces and displays that information on NNMi maps.

Forwarding Database (FDB) information can cause NNMi to establish wrong L2 Connections in the following cases:

- When the FDB is configured as cache and contains obsolete data.
- In network environments with hardware from a variety of vendors, each generating different and sometimes conflicting FDB data.

Optional: NNMi administrators can configure Discovery to ignore this FDB data for one Node Group.

Rediscover a Device

1. Perform a configuration poll of the device to confirm that you want to delete the device.
2. Delete the device.

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

Tune Discovery

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

- Filter by IP address range, 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 `nmnodelete.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 `nmseeddelete.ovpl` command.

Special discovery circumstances might be remedied by suppressing discovery protocol collections or VLAN-indexing. See ["Suppressing the Use of Discovery Protocols for Specific Nodes" on page 214](#) or ["Suppressing the Use of VLAN-indexing for Large Switches" on page 216](#) for more information.

Discovery Log File

To see what discovery classes are failing, look in the `nm.log` file for messages containing the keyword **Exception** for the classes beginning with the string `com.hp.ov.nms.disco`.

For information about log files, see ["NNMi Logging" on page 230](#).

Unnumbered Interfaces

NNMi enables you to discover and monitor unnumbered interfaces and the associated layer 2 connections, including those in a Global Network Management (GNM) environment.

If you are enabling layer 2 connectivity for unnumbered interfaces in a GNM environment, you must do so on both the regional managers and the global manager.

You can configure (enable and disable) layer 2 connectivity for unnumbered interfaces using NNMi's **Configuration > Discovery** workspace. See the NNMi Help for Administrators for more information.

Optionally, use the `nmunnumberedcfg.ovpl` command to configure unnumbered interface connectivity. See the `nmunnumberedcfg.ovpl` reference page, or the Linux manpage, for more information.

Note: Node Groups are not replicated between regional managers and the global manager.

You can use the `nmunnumberedcfg.ovpl` command to replicate configuration settings between a global manager and regional managers. This functionality lets you define Node Groups differently between the regional managers and the global manager. For example, you can define all routers at the global level and define only a subset of routers at each regional manager.

It is recommended that you have different configurations on the global manager than on the regional managers. For example, unless you are managing nodes directly from the global manager, there is no need to configure the optional subsets on the global manager because the data is only gathered at the regional manager.

Controlling Deletion of 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 control the deletion of unresponsive objects, perform the following steps:

1. In the **Configuration** workspace, click **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. Note that a value of zero (0) indicates that the objects should not be deleted.

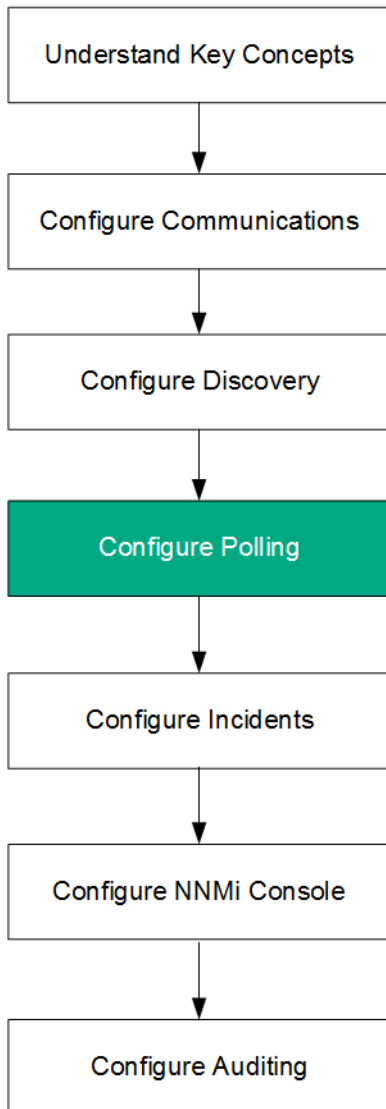
After the specified waiting period, the unresponsive objects are deleted from the database.

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 Administrators.

NNMi State Polling



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

This chapter contains the following topics:

- ["Concepts for State Polling" on the next page](#)
- ["Plan State Polling" on the next page](#)
- ["Configure State Polling" on page 83](#)
- ["Evaluate State Polling" on page 85](#)
- ["Tune State Polling" on page 87](#)

Concepts for State Polling

This section provides a brief overview of network monitoring, including the order that the State Poller uses to evaluate polling groups. After reading the information in this section, continue to ["Plan State Polling" below](#) for more specific information.

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

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

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

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 polling configuration, you should carefully consider how to set up interface groups and node groups for the State Poller service. If you are new to the concept of *groups*, see ["Node Groups and Interface Groups" on page 26](#), and ["Node Interface and Address Hierarchy" on page 29](#) for overview information.

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:

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

Plan State Polling

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

Polling Checklist

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

- What do I want NNMi to monitor?
- What are the logical groups for monitored items, based on object type, location, relative importance, or other criteria?
- How often should NNMi monitor each grouping?
- What data should be collected to capture information about the monitored item? This might include:
 - ICMP (ping) response
 - SNMP fault data
 - SNMP performance data if you have a license for one or more NNM Performance iSPIs
 - 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).

- What SNMP traps from my network devices should I send to NNMi?

Example polling configuration

To help you understand the polling configuration process, consider this example. Suppose that your network contains the latest proxy servers from ProximiT. You 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 ProximiT proxy servers. For more information on configuring discovery, see ["NNMi Discovery" on page 55](#).

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 should NNMi monitor each group?

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

4. What data should be collected?

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

3. What SNMP traps should be sent from my network devices to NNMi?

NNMi uses some SNMP traps to poll a 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:

- ["What Can NNMi Monitor?" on the next page](#)
- ["Planning Groups" on page 78](#)
- ["Planning Polling Intervals" on page 80](#)

- ["Deciding What Data to Collect" on page 81](#)
- ["Deciding What SNMP Traps to Send to NNMi" on page 82](#)

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)

Tip: 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 Administrators.

Tip: 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 Administrators.

For more information about monitoring, see the NNMi help.

Also see ["Extend Monitoring" on the next page](#)

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.

Tip: 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.

Interfaces to Unmonitored Nodes

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

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

- Discover the unmonitored node

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

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

- Poll the unconnected interface

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

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

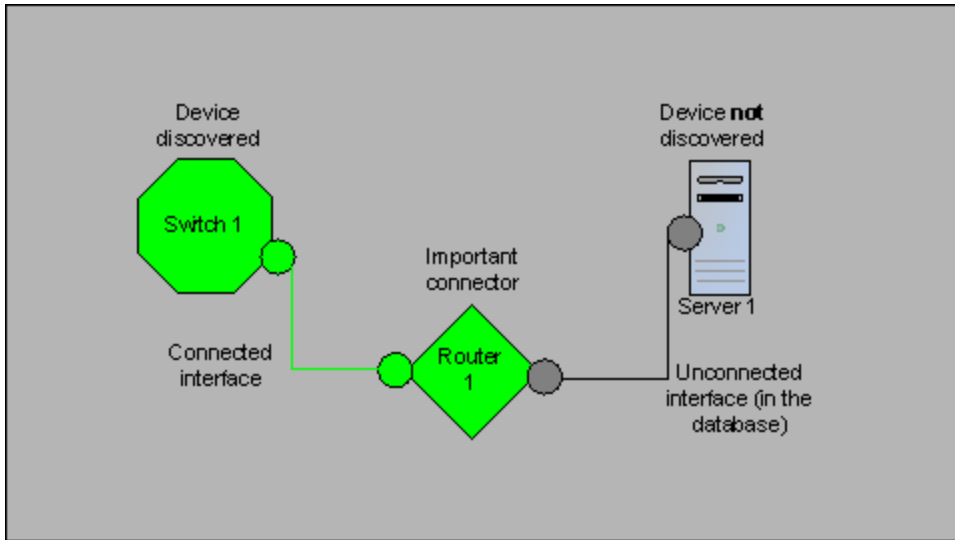
Extend Monitoring

You can extend the monitoring to include the following:

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

Note: NNMi defines an unconnected interface as an interface that is not connected to another device discovered by NNMi, as shown in the following diagram.

Unconnected Interface Example



- 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.

Planning Groups

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

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

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

Determine the criteria by which you want to group interfaces, 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?

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

Interface Groups

Based on your criteria, determine which Interface groups to create. Remember that interface groups are evaluated first (see ["Concepts for State Polling" on page 74](#)). Interface groups can reference node group membership, so you might end up configuring node groups before interface groups to implement your plan.

Preconfigured interface groups

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

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

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

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

- All interfaces on nodes in a node group are grouped regardless of IFTYPE; do not select any IFTYPES or attributes (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.

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

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

- Routers
- Networking Infrastructure Devices (such as switches or routers.)
- Microsoft Windows Systems
- Devices for which you do not have the SNMP community string

- **Important Nodes.** This is used internally by the Causal Engine to provide special handling for devices in the “shadow” of a connector failure. For more information, see *Node Groups As Predefined View Filters* in the NNMi help.
- **Virtual Machines**

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

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

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

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

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

Interaction with Device Profiles

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

As you configure node groups, you can use these derived attributes to categorize devices to apply monitoring settings. For example, you 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, Switches, will receive the configured settings for the node group.

Tip: 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.

Planning Polling Intervals

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

Tip: Shorter intervals help you become aware of network problems as soon as possible; however,

polling too many objects in too short an interval can cause a backlog in the State Poller. Find the best balance between resource utilization and intervals for your environment.

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.

Deciding What Data to Collect

The State Poller service uses polls to gather state information about the monitored devices in your network. Polling can be done using ICMP, 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. If you have the NNM iSPI Performance for Metrics installed, some values are used for performance measurement.

Web Polling

If Web Agents are configured (in addition to SNMP Agents), NNMi can use additional protocols. For example, **SOAP**¹ protocol for **VMware**² 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.

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

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

¹Simple Object Access Protocol

²VMware ESX and VMware ESXi software uses SOAP protocol to implement bare-metal hypervisors.

Deciding What 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

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

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

Tip: For more information about these SNMP Trap Incident configurations, from the NNMi console, navigate to the Configuration workspace and select **Incidents > SNMP Trap Configuration**.

Also see "[Use Discovery Hints from SNMP Traps](#)" on page 65.

Configure State Polling

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

Configure Interface Groups and Node Groups

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

Examples

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

1. Open **Configuration > Node Groups** and click *** New**.
2. Name the group **Proxy Servers** and check **Add to View Filter List**.
3. On the **Additional Filters** tab, select the **hostname** attribute, and select the equal (=) operator.
4. 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.

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

Configure Interface Monitoring

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

- Enabling or disabling fault monitoring
- Setting the fault polling interval
- Enabling or disabling performance polling if you have the NNM iSPI Performance for Metrics
- Setting the performance polling interval if you have the NNM iSPI Performance for Metrics
- Setting performance management thresholds if you have the NNM iSPI Performance for Metrics
- Selecting whether NNMi should 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.

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

Configure Node Monitoring

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

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

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

You might configure different settings for each node group.

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

Verify Default Settings

State Poller applies the settings from the **Default Settings** tab for any object that does not match a defined interface setting or node setting. Review the settings on this tab to ensure they match your environment at the

default level. For example, you would rarely poll all unconnected interfaces as a default setting.

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

Evaluate State Polling

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

Verify the Configuration for Network Monitoring

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

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

- ["Is the interface or node a member of the right group?" below](#)
- ["Which settings are being applied?" below](#)
- ["Which data is being collected?" on the next page](#)

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

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

- Node Groups
- Interface Groups

Follow the instructions in the help to show the members of the group. Keep in mind that an object can be a member of multiple groups, and that another group 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 under **Configuration > Device Profiles**.
3. Review the attribute requirements for the node group definition.

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

Which settings are being applied?

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

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

You might need to check **Actions > Communication Settings** for the object to ensure traffic has not been disabled.

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.

Note: If Web Agents are configured (in addition to SNMP Agents), NNMi can use additional protocols. For example, **SOAP**¹ protocol for **VMware**² 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.

Evaluate the Performance of Status Polling

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

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

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.

State Poller Health Information

Information	Description
Status	Overall status of the state poller service
Poll counters	<ul style="list-style-type: none"> • Collections requested • Collections completed • Collections in process • Collection request delays
Time to execute skips in last minute	<p>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.</p> <ul style="list-style-type: none"> • 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 <code>nm.?.?.log</code> file for messages for the classes beginning

¹Simple Object Access Protocol

²VMware ESX and VMware ESXi software uses SOAP protocol to implement bare-metal hypervisors.

State Poller Health Information, continued

Information	Description
	<p>with the string <code>com.hp.ov.nms.statepoller</code> to determine the targets for the skipped polls.</p> <ul style="list-style-type: none"> • 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 <code>ovjboss</code>.
Stale collections in last minute	<p>A stale collection is a collection that has not received a response from the polling engine for at least 10 minutes. A healthy system should never have any stale collections.</p> <ul style="list-style-type: none"> • What to watch for: If this value increases consistently, there is a problem with the polling engine. • Action to take: Look in the <code>nm.?.?.log</code> file for messages for the classes beginning with the string <code>com.hp.ov.nms.statepoller</code> to determine the targets for the stale collections. <ul style="list-style-type: none"> • If the stale collections are for a single target, unmanage the target until you can resolve the problem. • 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	<ul style="list-style-type: none"> • What to watch for: This value should be close to 0 most of the time. • Action to take: If this queue size is very large, <code>ovjboss</code> might be running out of memory.
State mapper queue duration	<ul style="list-style-type: none"> • What to watch for: This value should be close to 0 most of the time. • Action to take: If this queue duration is very large, then check the performance of the NNMI system and the NNMI database.
State updater queue duration	<ul style="list-style-type: none"> • What to watch for: This value should be close to 0 most of the time. • Action to take: If this queue size is very large, then check the performance of the NNMI system and the NNMI database.
State updater exceptions	<p>What to watch for: This value should be 0.</p>

Tune State Polling

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

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

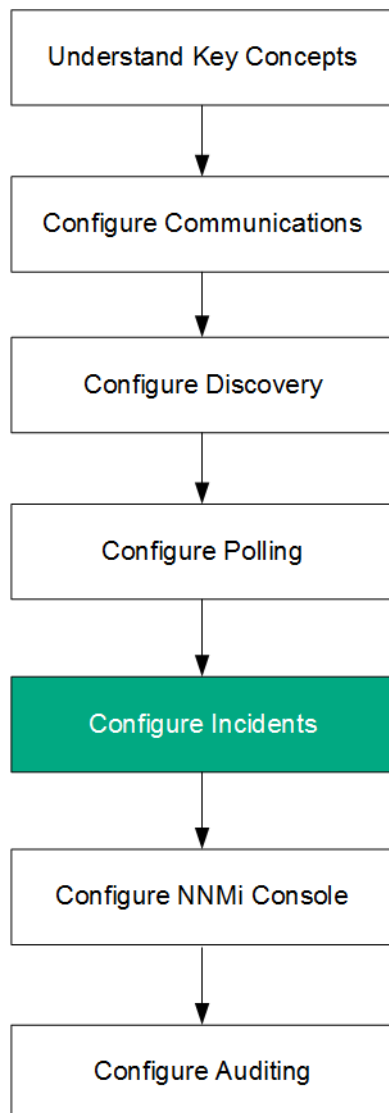
These variables are driven by your network management needs. If you are experiencing performance issues with status polling, consider the following configurations:

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

Regardless of the monitoring configuration, status polling is dependent on network responsiveness and 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 the Component Health monitoring has no effect on timeliness of polling. It simply gathers additional MIB objects at the schedule time. However, disabling Component Health monitoring might reduce the amount of memory used by the State Poller.

NNMi Incidents



HPE Network Node Manager i Software (NNMi) provides a large number of default incidents and correlations that filter incoming SNMP traps to provide a workable number of incidents in the NNMI console. This chapter provides information to help you fine tune network management by configuring the NNMI incidents. This chapter supplements the information in the NNMI help. For an introduction to NNMI incidents and for detailed information about how to configure incidents, see *Configuring Incidents* in the NNMI help.

This chapter contains the following topics:

- ["Concepts for Incidents" on the next page](#)
- ["Plan Incidents" on page 97](#)
- ["Configure Incidents" on page 98](#)
- ["Batch Load Incident Configurations" on page 100](#)
- ["Evaluate Incidents" on page 102](#)
- ["Tune Incidents" on page 102](#)

Concepts for Incidents

NNMi collects network status information from the following sources:

- The NNMi Causal Engine analyzes the health of your network and provides the ongoing health status reading for each device. The Causal Engine also extensively evaluates 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.
- Syslog messages from HPE ArcSight Logger integration.

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 in the NNMi console define the incident types that NNMi can create. If no incident configuration matches a received SNMP trap syslog message, that information is discarded. If the management mode of the source object is set to NOT MANAGED or OUT OF SERVICE in the NNMi database, or if the device is not monitored for fault polling, NNMi always discards the incoming trap.

Tip: `nmmtrapconfig.ovpl -dumpBlockList` outputs information about the current incident configuration, including SNMP traps that were not passed into the incident pipeline because of non-existent or disabled incident configurations.

Additionally, NNMi discards SNMP traps from network devices that are not in the NNMi topology. For information about changing this default behavior, see *Handle Unresolved Incoming Traps* in the NNMi help.

For more information, see the following:

- *About the Event Pipeline* in the NNMi help
- *The NNMi Causal Engine and Incidents* in the NNMi help
- *HPE Network Node Manager i Software Causal Analysis White Paper*, available from <http://h20230.www2.hp.com/selfsolve/manuals>

Incident Lifecycle

The following table describes the stages of an incident's lifecycle.

NNMi Incident Lifecycle

Lifecycle State	Description	State Set By	Incident Used By
none	The NNMi event pipeline receives input from all sources and creates incidents as needed.	not applicable	<ul style="list-style-type: none"> • 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.	NNMi	<ul style="list-style-type: none"> • NNMi

NNMi Incident Lifecycle, continued

Lifecycle State	Description	State Set By	Incident Used By
	The dampening interval can vary per incident type. For more information, see "Incident Suppression, Enrichment, and Dampening" on page 96.		
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.	<ul style="list-style-type: none"> Users Lifecycle transition actions Integrations that forward incidents
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	<ul style="list-style-type: none"> Users Lifecycle transition actions Integrations that forward incidents
Completed	Investigation of the problem indicated by the incident is complete, and a solution is in place. The problem that the incident identifies The network administrator defines the specific meaning of this state.	User	<ul style="list-style-type: none"> Users Lifecycle transition actions Integrations that forward incidents
Closed	Indicates that NNMi determined the problem reported by this Incident is no longer a problem. For example, when you remove an interface from a device, all incidents related to the interface are automatically closed.	User or NNMi	<ul style="list-style-type: none"> Users Lifecycle transition actions Integrations that forward incidents

Trap and Incident Forwarding

The following table summarizes the ways to forward traps and incidents from the NNMi management server to another destination. The text following the table compares the NNMi SNMP trap forwarding mechanism with the NNMi northbound interface SNMP trap forwarding mechanism.

Supported Ways to Forward Traps and NNMi Incidents

	NNMi Trap Forwarding	NNMi Northbound Interface Trap Forwarding	Global Network Management Trap Forwarding
What to forward	<ul style="list-style-type: none"> SNMP traps from network devices syslog messages from 	<ul style="list-style-type: none"> SNMP traps from network devices NNMi management 	<ul style="list-style-type: none"> SNMP traps from network devices syslog messages from

Supported Ways to Forward Traps and NNMi Incidents, continued

	NNMi Trap Forwarding	NNMi Northbound Interface Trap Forwarding	Global Network Management Trap Forwarding
	HPE ArcSight Logger	events <ul style="list-style-type: none"> • syslog messages from HPE ArcSight Logger 	HPE ArcSight Logger
Forwarding format	SNMPv1, v2c, or v3 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	Trap Forward Configuration in the Configuration workspace	HPOM, Northbound Interface , or Netcool in the Integration Module Configuration workspace	Forward to Global Managers tab on an SNMP Trap Configuration form or syslog configuration.
Notes		NNMi provides several integrations built on the NNMi northbound interface. <i>Also see the HPE Network Node Manager i Software—IBM Tivoli Netcool/OMNIbus Integration Guide and HPE Network Node Manager i Software—HPE Operations Manager Integration Guide.</i>	Forward the remote incidents that should be visible in the global manager incident views. Forwarded incidents participate in correlations on the global manager.
For more information	<i>Configuring Trap Forwarding</i> in the NNMi help	See the "NNMi Northbound Interface" chapter in the NNMi Deployment Reference.	<ul style="list-style-type: none"> • <i>Configure Forward to Global Manager Settings for an SNMP Trap Incident</i> in the NNMi help

Comparison: Forwarding Third-Party SNMP Traps to Another Application

If you want to forward the SNMP traps that NNMi receives from managed devices to another application, you can use either of the following approaches:

- Use the NNMi SNMP trap forwarding mechanism. For information about how to configure NNMi SNMP trap forwarding, see *Configuring Trap Forwarding* in the NNMi help.
- Use the NNMi northbound interface SNMP trap forwarding mechanism. For information about configuring the NNMi northbound interface to forward received SNMP traps, the NNMi Northbound Interface chapter in the *NNMiIntegration Reference*.

The approach to trap identification by the receiving application varies with the SNMP trap forwarding mechanism:

- *Windows (all) and Linux without original trap forwarding*

This description applies to the default and 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 `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 (`ipv4(1)`, `ipv6(2)`) value in the `originIPAddress` varbind. The rule might also need to convert the `originIPAddress` value to a display string.

For more information about the varbinds that NNMi adds to forwarded traps, see *Trap Varbinds Provided by NNMi* in the 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 the traps in the same format as 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 should work without modification.

For more information, see the original trap forwarding option in *Trap Forwarding Destination Form* in the NNMi help.

- *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 `IncidentNodeHostname (1.3.6.1.4.1.11.2.17.19.2.2.21)` and `IncidentNodeMgmtAddr (1.3.6.1.4.1.11.2.17.19.2.2.24)` varbinds identify the original source object.

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)
- (NNM iSPI Performance for Metrics) All MIB variables used in threshold monitoring

NNMi requires that the following management information base (MIB) files, or the traps defined in the MIB files, be loaded into the NNMi database:

- All SNMP traps that you want to forward to a northbound destination
- (NNM iSPI NET) All MIB variables accessed from Trap Analytics reports

Tip: NNMi provides a `README.txt` file that lists those MIBs that are currently not supported. The `README.txt` file is located in the following directory:

- *Windows:* `%NmInstallDir%\misc\nnm\snmp-mibs`
- *Linux:* `$NmInstallDir/misc/nnm/snmp-mibs`

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 menu items on the Actions menu for an incident view or form.

To determine which CIAs NNMi adds for any given incident, open a sample incident from an incident view, and look at the information on the Custom Attributes tab.

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 following table 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.

Custom Incident Attributes for a Closed Incident

Name	Description
<code>cia.reasonClosed</code>	<p>The reason that NNMi cancelled or closed the incident. This reason is also the conclusion name, for example <code>NodeUp</code> or <code>InterfaceUp</code>.</p> <p>If this field is not set, an NNMi console user closed the incident.</p> <p>To determine the NNMi expected values of the <code>cia.reasonClosed</code> CIA, see <i>How NNMi Closes Incidents</i> in the NNMi help.</p>
<code>cia.incidentDurationMs</code>	<p>The duration, in milliseconds, of the outage, as measured by NNMi from when the status goes down and comes back up. This value is the difference of the <code>cia.timeIncidentDetectedMs</code> and</p>

Custom Incident Attributes for a Closed Incident, continued

Name	Description
	cia.timeIncidentResolvedMs CIAs. It is a more accurate measurement than comparing the timestamps of down and up incidents.
cia.timeIncidentDetectedMs	The timestamp, in milliseconds, when the NNMi Causal Engine first detected the problem.
cia.timeIncidentResolvedMs	The timestamp, in milliseconds, when the NNMi Causal Engine detected that the problem has been resolved.

NNMi adds the CIAs listed in the previous table to most primary and secondary root cause incidents. For example, a NodeDown incident can have InterfaceDown and AddressDown 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 the previous table 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

Incident Reduction

NNMi provides the following customizable correlations for reducing the number of incidents that network operators see in the NNMi console:

- Pairwise correlation—One incident cancels another incident.
- 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.

Incident Suppression, Enrichment, and Dampening

NNMi provides a rich feature set for getting the most value from incidents. For each incident type, you can specifically define when an incident is of interest with the following incident configuration options:

- **Suppression**—When an incident matches the suppression configuration, that incident does not appear in the NNMi console incident views. Incident suppression is useful for incidents (for example, SNMPLinkDown traps) that are important for some nodes (for example routers and switches) but not 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 for that incident for the duration of the dampen 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 in 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.

For each incident configuration area (suppression, enrichment, and dampening), NNMi uses the following procedure to determine the behavior of a specific incident:

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, continue with [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, continue with [step 3](#).
3. Carry out the behavior defined in the default settings, if any.

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 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.

Plan Incidents

Make decisions in the following areas:

- ["Which Device Traps Should NNMi Process?" below](#)
- ["Which Incidents Should NNMi Display?" below](#)
- ["How Should NNMi Respond to Incidents?" below](#)
- ["Should NNMi Forward Traps to Another Event Receiver?" on the next page](#)

Which Device Traps Should NNMi Process?

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. If the MIB contains TRAP-TYPE or NOTIFICATION-TYPE macros, you can create skeleton incident configurations for the traps defined in the MIB.

Decide whether you want to see traps from devices that are not in the NNMi topology.

Which Incidents Should NNMi Display?

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 through deduplication, rate configuration, and pairwise correlation.

See the NNMi Help for Administrators for more information.

How Should NNMi Respond to Incidents?

What actions (for example, sending an email message to a network operator) should NNMi take when certain incidents occur? At what lifecycle state should each action run?

See the NNMi Help for Administrators for more information.

Should NNMi Forward Traps to Another Event Receiver?

If your environment includes a third-party trap consolidator, decide whether to use the NNMi SNMP trap forwarding mechanism with the NNMi northbound interface SNMP trap forwarding mechanism.

If you choose the NNMi northbound interface SNMP trap forwarding mechanism, load the MIBs for all traps that NNMi will forward to the event receiver.

Configure Incidents

This section lists configuration tips and provides some configuration examples. After reading the information in this section, see *Configuring Incidents* in the 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.

- Configure the incident types that you planned. If possible, start with the skeleton incident configurations from the traps defined in the MIB.
- Load any MIBs that are required for trap forwarding.
- Verify that devices are configured to send traps to the NNMi management server.

Configuring Incident Suppression, Enrichment, and Dampening

While configuring incident suppression, enrichment, and dampening, note the following:

- 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.

Configuring Lifecycle Transition Actions

While configuring lifecycle transition actions, note the following:

- By default, NNMi runs actions in the following location:
 - *Windows:* %NnmDataDir%\shared\nnm\actions
 - *Linux:* \$NnmDataDir/shared/nnm/actions

If an action is not in this location, specify the absolute path to the action in the **Command** field of the **Lifecycle Transition Action** form.

Note: 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 information about the NNMi information that you can pass to an action, see *Valid Parameters for Configuring Incident Actions* in the NNMi help.

Configuring Trap Logs

NNMi provides the ability to log all incoming SNMP traps into a log file (either a text file or a CSV file). Traps are logged to the following location:

- *Windows*: %NnmDataDir%\nnm\log
- *Linux*: \$NnmDataDir/nnm/log

Trap log files can be configured using the `nnmtrapconfig.ovpl` script. The following format choices are available:

- CSV (default) – Traps are logged in the CSV format (`trap.csv`).
- TXT – Traps are logged in the TXT format (`trap.log`).
- BOTH – Traps are logged in both CSV and TXT (2 log files).
- OFF – No traps are logged.

For example, to specify that traps get logged into BOTH modes, you would use the following command:

```
nnmtrapconfig.ovpl -setProp trapLoggingMode BOTH -persist
```

Note that the `-persist` argument causes all trap server properties to remain in effect even after the trap service is restarted. If you do not use the `-persist` argument, all trap server properties will be in effect only until the service is stopped.

Traps are written to a rolling file. After the log file size reaches the defined maximum limit (as defined using the `nnmtrapconfig.ovpl` script), the file is renamed to `trap.<format>.old`. Any existing file is replaced.

See the `nnmtrapconfig.ovpl` reference page, or the Linux manpage, for more information. See also *Configure Trap Logging* in the NNMi help.

Configuring Incident Logging

You can configure incident logging so that incoming incident information is written to the `incident.log` file. This feature is useful when you want to track and archive your incident history.

Configure and enable incident logging by navigating to the **Incident Logging Configuration** tab in the **Incident Configuration** area of the **Configuration** workspace, and configuring the settings. For more information, see the NNMi help.

Configuring Trap Server Properties

You can set trap server properties (`nnmtrapserver.properties`) by using the `nnmtrapconfig.ovpl` script.

Note: Although an `nnmtrapserver.properties` file exists, do not edit this file directly; use the `nnmtrapconfig.ovpl` script to modify the file.

The following table shows the default values for trap server properties.

Trap Server Properties and Default Values

Trap Server Property	Default Value
com.hp.ov.nms.trapd.udpPort	162
com.hp.ov.nms.trapd.rmiPort	1097
com.hp.ov.nms.trapd.trapInterface	all interfaces
com.hp.ov.nms.trapd.recvSocketBufSize	2048 kilobytes
com.hp.ov.nms.trapd.pipeline.qSize	50000 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	300000 traps
com.hp.ov.nms.trapd.discohint.cacheSize	5000 entries
com.hp.ov.nms.trapd.discohint.cacheEntryTimeout	3600 milliseconds

See the *nmtrapconfig.ovpl* reference page or the Linux manpage for more information.

Batch Load Incident Configurations

Use the following two scripts in conjunction with batch loading of incident configurations:
nmincidentcfgdump.ovpl and *nmincidentcfgload.ovpl*.

Generating an Incident Configuration File with `nnmincidentcfgdump.ovpl`

The NNMi `nnmincidentcfgdump.ovpl` script provides a way for you to create or update an Incident Configuration to subsequently load into the NNMi database using the `nnmincidentcfgload.ovpl` script. The file is generated in a non-xml format.

You can edit the file using the format descriptions provided in the following directory:

Windows: %NnmInstallDir%/examples/nnm/incidentcfg

Linux: /opt/OV/examples/nnm/incidentcfg

To generate a file of your Incident Configurations, use the following example syntax:

```
nnmincidentcfgdump.ovpl -dump <file_name> -u <NNMiadminUsername>
-p <NNMiadminPassword>
```

See the `nnmincidentcfgdump.ovpl` reference page or the Linux manpage for more information.

Note: When making file changes under High Availability (HA), you must make the changes on both nodes in the cluster. For NNMi using HA configurations, if the change requires you to stop and restart the NNMi management server, you must put the nodes in maintenance mode before running the `ovstop` and `ovstart` commands.

Loading Incident Configurations with `nnmincidentcfgload.ovpl`

The NNMi `nnmincidentcfgload.ovpl` script provides a way for you to load Incident Configurations into the NNMi database from a formatted configuration file.

Tip: Use the `nnmincidentcfgdump.ovpl` script to create a configuration file of existing Incident Configurations in a non-xml format. You can then edit this file if desired before loading them into the NNMi database.

See the following directory for the required format:

Windows: %NnmInstallDir%\examples\nnm\incidentcfg

Linux: /opt/OV/examples/nnm/incidentcfg

To validate an Incident Configuration file before it is loaded into the NNMi database, use the following example syntax:

```
nnmincidentcfgload.ovpl -validate <file_name> -u <NNMiadminUsername>
-p <NNMiadminPassword>
```

To load Incident Configurations, use the following example syntax:

```
nnmincidentcfgload.ovpl -load <file_name> -u <NNMiadminUsername>
-p <NNMiadminPassword>
```

Note the following:

- NNMi updates all configurations that have matching names or other matching key identifiers.

Caution: NNMi also overwrites the values of any codes associated with these configurations (for example, incident Family).

- NNMi adds all incident configurations with key identifiers that do not exist in the NNMi database.
- NNMi does not change existing incident configurations with key identifiers that do not match any in the exported file.
- NNMi resolves Universally Unique Object Identifiers (UUIDs) if they are not provided in the configuration file.
- If NNMi is unable to resolve a UUID, a UUID is created.

Note: When making file changes under High Availability (HA), you must make the changes on both nodes in the cluster. For NNMi using HA configurations, if the change requires you to stop and restart the NNMi management server, you must put the nodes in maintenance mode before running the ovstop and ovstart commands.

See the `nnmincidentcfgload.ovpl` reference page or the Linux manpage for more information.

Evaluate 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, verify the configuration of the firewall on the NNMi management server.

Note: 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 **Actions > Incident Configuration Reports** menu 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 in the NNMi console.

Tune Incidents

Reduce the number of incidents in the NNMi console incident views. Use any of the following methods:

- Disable the incident configuration for any incident types that are not needed in 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 most incoming traps from these nodes and their interfaces.
- Set NNMi to not monitor some network objects. NNMi discards most incoming traps from the 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.

Enabling and Configuring Incidents for Undefined Traps

NNMi drops undefined traps silently by default. As of NNMi 9.01, NNMi can identify any undefined SNMP traps that might be dropped.

Note: If you have NNM iSPI NET or NNMi Premium licensed on the NNMi management server, use the Total Traps Received (by OID) report to research the dropped SNMP traps. See *Analyze Trap Information (NNM iSPI NET)* in the NNMi help for more information.

If you do not have NNM iSPI NET or NNMi Premium licensed on the NNMi management server, and want to see the missing traps as an incident, configure the Undefined SNMP Trap incident as follows:

1. Edit the following file:
 - *Windows:* %NNM_PROPS%\nms-jboss.properties
 - *Linux:* \$NNM_PROPS/nms-jboss.properties
2. Look for a section in the file that resembles the following line:


```
#!com.hp.nnm.events.allowUndefinedTraps=false
```

 Change this line as follows:


```
com.hp.nnm.events.allowUndefinedTraps=true
```
3. *Optional.* Specify the incident severity using the values explained within the nms-jboss.properties file. Look for a section in the file that resembles the following line:


```
#!com.hp.nnm.events.undefinedTrapsSeverity=NORMAL
```

 Change this line as follows, substituting a defined severity value for *YourSpecifiedSeverity*.

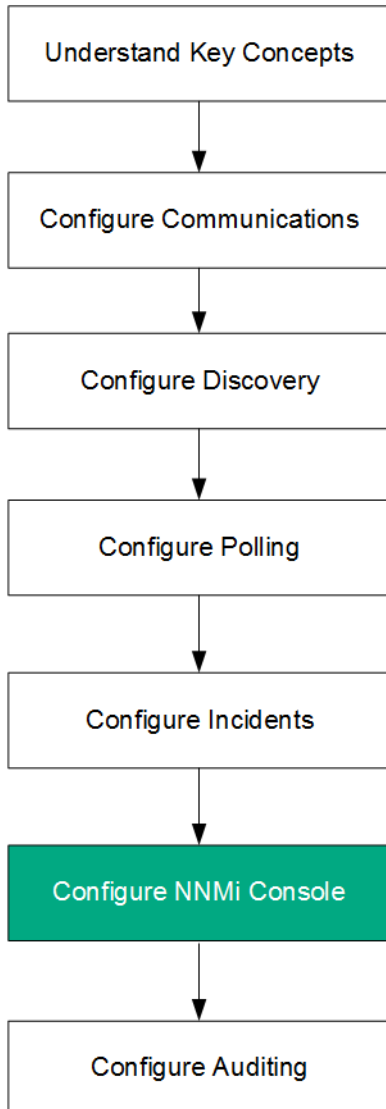

```
com.hp.nnm.events.undefinedTrapsSeverity=YourSpecifiedSeverity
```
4. *Optional.* Specify the incident nature using the values explained within the nms-jboss.properties file. Look for a section in the file that resembles the following:


```
#!com.hp.nnm.events.undefinedTrapsNature=INFO
```

 Change this line as follows, substituting a defined nature value for *YourSpecifiedNature*.


```
com.hp.nnm.events.undefinedTrapsNature=YourSpecifiedNature
```
5. Restart the NNMi management server.
 - a. Run the ovstop command on the NNMi management server.
 - b. Run the ovstart command on the NNMi management server.
6. Review the list of undefined traps and create new incident configurations for those traps that you want to control. Enable the new incident if you want NNMi to display it and disable the new incident if you want NNMi to ignore it. See *Configuring SNMP Trap Incidents* in the NNMi help for more information.

Configure NNMi Console



Use the information in this chapter to understand how to use the NNMi console to configure NNMi to function in specific ways.

This chapter contains the following topics:

- ["Reduce the Maximum Number of Nodes Displayed in a Network Overview Map" on the next page](#)
- ["Reduce the Number of Displayed Nodes on a Node Group Map" on the next page](#)
- ["Configure Gauges in the Analysis Pane" on page 106](#)
- ["Configuring Map Label Scale Size and Borders" on page 108](#)
- ["Configuring Auto-Collapse Thresholds for Loom and Wheel Diagrams" on page 109](#)
- ["Customize Device Profile Icons" on page 110](#)
- ["Configure a Table View's Refresh Rate" on page 110](#)

Reduce the Maximum Number of Nodes Displayed in a Network Overview Map

The **Network Overview** map displays a map containing up to 250 of the most highly connected nodes in the layer 3 network. If this map contains too many nodes, the map might respond slowly when moving nodes or become too complex for practical viewing.

You can increase or reduce the maximum number of nodes displayed in the **Network Overview** map by editing the Maximum Number of Displayed Nodes attribute on the **Default Map Settings** tab on the User Interface Configuration form.

You can also increase or reduce the maximum number of nodes displayed in the **Network Overview** map by performing the steps shown in the following example.

For example, to change the maximum number of nodes displayed in the **Network Overview** map from 250 to 100, follow these steps:

1. Edit the following file:
 - *Windows:* %NNM_PROPS%\nms-ui.properties
 - *Linux:* \$NNM_PROPS/nms-ui.properties

2. Look for text similar to following line:

```
#!com.hp.nnm.ui.networkOverviewMaxNodes = 250
```

Change the line as follows:

```
com.hp.nnm.ui.networkOverviewMaxNodes = 100
```

Note: Make sure to remove the **#!** characters located at the beginning of the line.

3. Save your changes.

Reduce the Number of Displayed Nodes on a Node Group Map

If you configure a node group map to contain hundreds of nodes, the map showing the 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.

Note: Using the zoom feature might slow the NNMi console performance when displaying maps.

To limit the number of displayed nodes, displayed end points, or both, follow these steps:

1. In the NNMi console, click **Configuration**.
2. Click **User Interface Configuration**.
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**.

See *Define Default Map Settings* in the NNMi help for more information.

Configure 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.

Limit 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 =
```

Tip: A higher number of gauges affects performance when the analysis pane is displayed. A fewer number of gauges results in larger size gauges.

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 =
```

Tip: 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.

Eliminate 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.

Control the Order of Displayed Node Gauges

To control the order in which node gauges are displayed, edit the following line:

```
com.hp.nnm.ui.analysisGaugeNodeComponentKeys =
```

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.

Control 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 =
```

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.

Control 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.

Understand 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.

Troubleshoot Gauge Problems

This section includes information for troubleshooting the following gauge problem:

- ["Too Many Gauges Are Displayed" below](#)

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 ["Limit the Number of Gauges Displayed" on page 106](#) for more information.
- Use the `analysisGaugeNoDisplayKeyPatterns` property to remove the gauges that are not wanted
See ["Eliminate Gauges from the Display" on the previous page](#) for more information.

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.

Tip: 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.

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.maxLabelScaleAdjust	1.0	Adjusts the largest relative scale factor that can be used to determine the difference in size between nodes or ports and their labels.
!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: To implement your changes, re-open or change your map view.

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.

Auto-Collapse Thresholds for Wheel and Loom

Property	Description
<code>com.hp.nnm.ui.wheelAutoCollapseThreshold</code>	Use this property to specify the number of labels required around the perimeter before the Wheel Diagram automatically collapses.
<code>com.hp.nnm.ui.loomAutoCollapseThreshold</code>	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:

- Edit the following file:
 - Windows: `%NNM_PROPS\nms-ui.properties`
 - Linux: `$NNM_PROPS/nms-ui.properties`
- 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.

Disable the Analysis Pane

NNMI allows you to disable the analysis pane from the NNMI console by performing the following steps:

1. Edit the following file:
 - Windows: %NNM_PROPS%\nms-ui.properties
 - UNIX: \$NNM_PROPS/nms-ui.properties
2. Append the following text to the end of the file:


```
# Disables the analysis pane from being shown by default.
# The analysis pane can still be shown by toggling it open
# or using the "Show Analysis Pane" menu item.
# com.hp.nnm.ui.analysisPaneDisabled = true
```
3. Uncomment the property (the last line) to disable the analysis pane.
4. Save your changes.

Customize Device Profile Icons

NNMI enables you to customize icons associated with a Device Profile or specific Nodes. These icons appear in table views, menu items, and as foreground images on an NNMI topology map.

You can customize one or many icons using the `nnmicons.ovpl` command. For more information, see the *nnmicons.ovpl* reference page, or the Linux manpage.

See also the NNMI Help for Administrators.

Configure a Table View's Refresh Rate

NNMI enables an NNMI administrator to override the default refresh rate for a table view in the NNMI console.

Note: The minimum recommended refresh rate is 30 seconds. Setting the refresh rate less than 30 seconds can degrade performance.

To override the default refresh rate for an NNMI table view, complete the following steps:

1. Edit the following file:


```
Windows: %NMS-PROPS%\nms-ui.properties
Linux: $NNM_PROPS/nms-ui.properties
```
2. Determine the `viewInfoId` URL parameter of the view that has the refresh rate you want to change:
 - a. Open the view that has the refresh rate you want to change.
 - b. Click **Show View in New Window**.
 - c. Note the `viewInfoId` URL parameter. For example, **viewInfoId=allIncidentsTableView**.
3. Using the following format add a line to `nms-ui.properties` to specify the view and its refresh rate in

seconds:

```
com.hp.ov.nms.ui.refreshViewSecs.VIEWKEYWORD = SECS
```

Note the following:

- **VIEWKEYWORD** is the viewInfold URL parameter of the view.
- **SECS** is the refresh rate in number of seconds.
- Ensure 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`:

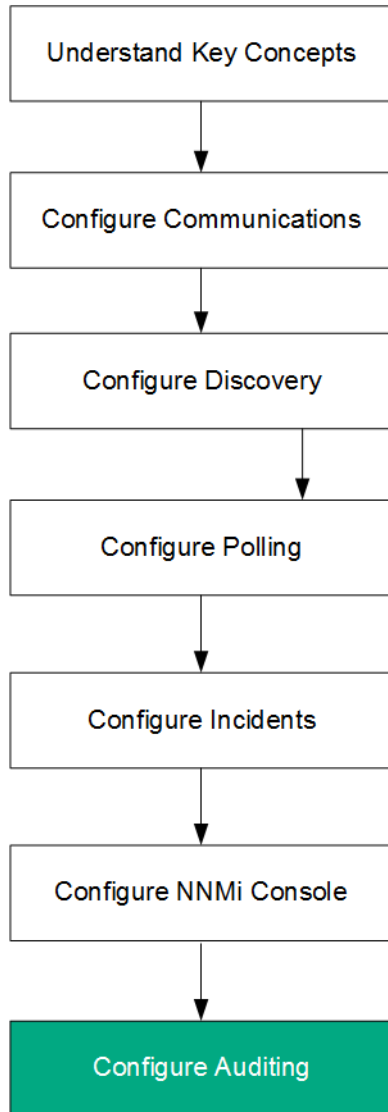
```
com.hp.ov.nms.ui.refreshViewSecs.allIncidentsTableView = 120
```

4. Save your changes.

To see the new refresh rate, open a different view and then return to the view that has the refresh rate you just configured.

Note: 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 ["Maintenance Mode" on page 177](#) for more information.

NNMi Auditing



By default, NNMi audits user actions that result in changes to the NNMi database. These kinds of user actions include, but are not limited to, the following:

- Changes to NNMi topology objects (for example, nodes, node groups, interfaces, and interface groups) . Examples include creating or deleting Node Groups or Interface Groups, and changing filters or membership in a Node Groups or Interface Groups.
- Changes to incident lifecycle information. Examples include changing an incident's owner or state.
- Changes to user and access information. Example include changing passwords, adding or deleting a user account or user group, and creating tenants.
- Configuration changes made using the NNMi console **Configuration** workspace or a command line tool. Example include modifications to SNMP settings, discovery settings, and monitoring configuration.
- User actions from the NNMi console **Actions** menu. Examples include Configuration Poll and Status Poll.

See "[About the NNMi Audit Log File](#)" on page 118 for examples of the type of information written to audit logs

Note: By default, the following actions or changes are NOT included in the audit log:

- Actions performed by the **system** user
- Automatically performed by NNMi are not included in the audit log. To change this default behavior, see ["Configure the Actions Included in the NNMi Audit Log File" on page 117](#)

Note the following:

- NNMi auditing is enabled by default.
- Audit information is written to one log file per day.
- The audit log files reside in the following directory:

Tip: As an NNMi administrator you can also view the most current audit log from the NNMi console **Tools > NNMi Audit Log** menu option.

Windows: %NnmDataDir%\nmsas\NNM\log\audit-<date>.log

Linux: \$NnmDataDir/nmsas/NNM/log/audit-<date>.log

• **Example Log Entries:**

User Action:

```
2014-10-26T22:00:21.305 admin 10.12.203.55 ACTION "" com.hp.nnm.ui.actions.configpoll Node
4295011152 cisco4k1 "" "" ""
```

Model Updates:

```
2014-04-30T01:20:25.301 joe.operator 10.12.203.55 MODEL abb44ddb-ae52-40d9-855f-
f6ab0ab899e1 UPDATE Node 151434 172.20.12.7 managementMode MANAGED NOTMANAGED
```

```
2014-04-15T01:55:48.574 admin "" MODEL 4654e06c-5c1f-4955-bf82-e317dcbf38f3 CREATE
Account 56647 op1 name "" op1
```

Each record in the audit log includes the following kinds of information:

Audit Log

Field	Description
Timestamp	When the audit record is created. In ISO-8601 format without a timezone (local time).
Username	The logged in username associated with the change.
Remote Address	For changes made via the NNMi Console this will be the address of the client system: <ul style="list-style-type: none"> • The remote address of the client if applicable. • "" (indicates not applicable).
Record Type	The category describing the type of change: <ul style="list-style-type: none"> • ACTION – An action run by the user. • ACCESS_DENIED – A security check was performed and the user was denied access to the specified action.

Audit Log, continued

Field	Description
	<ul style="list-style-type: none"> • MODEL – A change to an object in the NNMi topology or configuration made by the user. • MESSAGE - Log messages about the system rather than auditing of a user action. For example, the following series of messages might be logged when auditing has successfully begun and is subsequently stopped: 2015-08-24T22:37:01.012 system "" MESSAGE "Auditing started" 2015-08-24T22:37:01.014 system "" MESSAGE "Reloaded auditing configuration; auditing is enabled" 2015-08-24T22:37:01.015 system "" MESSAGE "Audit service initialized successfully" 2015-08-24T22:59:08.194 system "" MESSAGE "Audit service shutting down" 2015-08-24T22:59:08.195 system "" MESSAGE "Auditing stopped" • TX – Used to indicate transaction boundaries for very large changes. If a change has a very large number of entries then it is written progressively as changes are made and these entries will indicate if the transaction commits or rolls back.
Transaction ID	<p>Used to correlate multiple entries into a single transaction. Populated for all MODEL entries:</p> <ul style="list-style-type: none"> • ID • "" (indicates not applicable).
Operation / Action	<p>The specific operation or action associated with the entry.</p> <ul style="list-style-type: none"> • "" (means no action performed) <p>For MODEL record types:</p> <ul style="list-style-type: none"> • CREATE – Creating an entry in the NNMi database. • UPDATE – Updating an entry in the NNMi database. • DELETE – Deleting an entry in the NNMi database. <p>For TX record types:</p> <ul style="list-style-type: none"> • BEGIN – Records the start of a transaction. A matching COMMIT or ROLLBACK should appear later in the audit log to indicate the outcome of the transaction and all changes made within it. • COMMIT – The transaction committed and so all entries associated with that transaction in the audit log have been applied. • ROLLBACK – The transaction rolled back and so all entries associated with that transaction in the audit log were NOT applied.

Audit Log, continued

Field	Description
	For ACTION record types this entry contains a code indicating which action was performed by the user.
Target Object Type	When the record pertains to a type of object in NNMi this entry lists that type: <ul style="list-style-type: none"> For example, "Account" for a change to a user account. "" (if not applicable)
Additional meta data available for the object or action (if applicable):	
Target Object ID	When the record pertains to a specific object in NNMi this entry lists the unique ID of that object. "" (if not applicable)
Target Object Name	When this record pertains to a specific object in NNMi this entry lists a user-friendly name or label of that object (where available). "" (if not applicable)
Field Name	When this record pertains to a specific field on an object this identifies the field that was changed. For example "password" might be the field if the object type was "Account". "" (if not applicable)
Field Previous Value	When this record pertains to a specific change to a field on an object this entry lists the previous value of the field. <div style="background-color: #e0e0e0; padding: 5px;">Note: Sensitive information such as passwords values are displayed as asterisks, for example: password *****</div> <p>Create operations will have an empty value ("") in this position. Delete operations will have the value before delete in this position. "" (if not applicable)</p>
Field New Value	When this record pertains to a specific change to a field on an object this entry lists the new value of the field. <div style="background-color: #e0e0e0; padding: 5px;">Note: Sensitive information such as passwords values are displayed as asterisks, for example: password *****</div> <p>Create operations will have the initial value in this position. Delete operations will have an empty value ("") in this position. "" (if not applicable)</p>

See ["About the NNMi Audit Log File"](#) on page 118 for example log file entries.

- NNMi retains each audit log file for 14 days

As an NNMi administrator, you can configure the following:

- ["Disable Auditing" below](#)
- ["Specify the Number of Days to Retain NNMi Audit Logs" below](#)
- ["Configure the Actions Included in the NNMi Audit Log File" on the next page](#)

Disable Auditing

NNMi auditing is enabled by default.

To disable NNMi auditing:

1. Open the following configuration file:

Windows

```
%NnmDataDir%\nmsas\NNM\conf\nms-audit-config.xml
```

Linux

```
$NnmDataDir/nmsas/NNM/conf/nms-audit-config.xml
```

2. Locate the text block containing the following:

```
<enabled>true</enabled>
```

3. Change true to false:

```
<enabled>>false</enabled>
```

4. Save your changes.

5. Restart the NNMi management server:

Run the `ovstop` command on the NNMi management server.

Run the `ovstart` command on the NNMi management server.

Specify the Number of Days to Retain NNMi Audit Logs

By default, NNMi retains each archived audit log file, one per day, for 14 days.

To change the number of days that NNMi retains archived audit log file:

Note: This number does not affect the current day's audit log file.

1. Open the following configuration file:

Windows

```
%NnmDataDir%\nmsas\NNM\conf\nms-audit-config.xml
```

Linux

```
$NnmDataDir/nmsas/NNM/conf/nms-audit-config.xml
```

2. Locate the text block containing the following:

```
<retain>14</retain>
```

3. Modify the line to include the number of days NNMi should retain each audit log file. For example, to change the number of days to one week, enter:

```
<retain>7</retain>
```

In response, NNMi retains the following:

- the current audit log
 - one audit log per day for 7 additional days
4. Save your changes.
 5. Restart the NNMi management server:
 - Run the `ovstop` command on the NNMi management server.
 - Run the `ovstart` command on the NNMi management server

Configure the Actions Included in the NNMi Audit Log File

By default, NNMi audits user actions that result in changes to the NNMi database. These kinds of user actions include, but are not limited to, the following:

- Changes to NNMi topology objects (for example, nodes, node groups, interfaces, and interface groups). Examples include creating or deleting Node Groups or Interface Groups, and changing filters or membership in a Node Groups or Interface Groups.
- Changes to incident lifecycle information. Examples include changing an incident's owner or state.
- Changes to user and access information. Example include changing passwords, adding or deleting a user account or user group, and creating tenants.
- Configuration changes made using the NNMi console **Configuration** workspace or a command line tool. Example include modifications to SNMP settings, discovery settings, and monitoring configuration.
- User actions from the NNMi console **Actions** menu. Examples include Configuration Poll and Status Poll.

See ["About the NNMi Audit Log File" on the next page](#) for examples of the type of information written to audit logs

After you examine an NNMi audit log file, you might find that you want to include or exclude auditing for a particular action, entity or field. See [step 3](#) for examples.

Tip: In each audit log message, the `<action_name>` immediately precedes the `<entity_name>`. The field name appears after the `<entity_name>`. Here is an example message, with the action (UPDATE), entity (Node), and field name (managementMode) in bold:

```
2014-04-30T01:20:25.301 joe.operator 10.12.203.55 MODEL abb44ddb-ae52-40d9-855f-f6ab0ab899e1
UPDATE Node 151434 172.20.12.7 managementMode MANAGED NOTMANAGED
```

To change the information included in an NNMi audit log:

1. Open the following configuration file:
 - Windows


```
%NnmDataDir%\nmsas\NNM\conf\nms-audit-config.xml
```
 - Linux


```
$NnmDataDir/nmsas/NNM/conf/nms-audit-config.xml
```
2. Locate the text block containing the following:

```
<rules>
<!-- define custom audit rules here. Any rules here will override system defaults -->
</rules>
```

3. Modify the rules as follows:

- To exclude a single message in the audit log, use the following syntax:

```
<exclude entity="<entity_name>" field="<field_name>" action="<action_name>" />
```

The following example excludes this example audit log message:

```
2014-04-30T01:20:25.301 joe.operator 10.12.203.55 MODEL abb44ddb-ae52-40d9-855f-
f6ab0ab899e1 UPDATE Node 151434 172.20.12.7 managementMode MANAGED
NOTMANAGED
```

```
<exclude entity="Node" field="managementMode" action="UPDATE" />
```

- To exclude from the audit log all actions to an entity, use the following syntax:

```
<exclude entity="<entity_name>" />
```

The following example excludes from the audit log all update operations to nodes.

```
<exclude entity="Node" />
```

- To exclude a specified action to an entity, use the following syntax:

```
<exclude entity="<entity_name>" action="<action_name>" />
```

The following example excludes from the audit log all update operations to nodes.

```
<exclude entity="Node" action="UPDATE" />
```

The following example excludes from the audit log all delete operations to nodes:

```
<exclude entity="Node" action="DELETE" />
```

- To exclude from the audit log all actions to a specified field on any object, use the following syntax:

```
<exclude field="<field_name>" />
```

The following example excludes from the audit log all updates to the managementMode field on any object:

```
<exclude field="managementMode" action="UPDATE" />
```

4. Restart the NNMi management server:

Run the `ovstop` command on the NNMi management server.

Run the `ovstart` command on the NNMi management server

About the NNMi Audit Log File

This section provides examples of the types of information you will find in audit log files.

- Example audit log entry generated after changing a node's Security Group

The following is an example log entry that was generated when the Security Group of the node named **mimcisco3** was changed from **Default Security Group** to **testgrp**.

2014-04-15T01:56:54.979 admin "" MODEL 5fd8ed33-e671-494e-ab25-06d293347c4f **UPDATE Node 50281 mimcisco3 securityGroup "138/Default Security Group" 56651/testgrp**

- Example audit log entry generated when a User Account was created:

The following are example log entries that were generated when an account for user op1 was created:

2014-04-15T01:55:48.574 admin "" MODEL 4654e06c-5c1f-4955-bf82-e317dcbf38f3 CREATE Account 56647 **op1** alg "" SHA-256

2014-04-15T01:55:48.574 admin "" MODEL 4654e06c-5c1f-4955-bf82-e317dcbf38f3 CREATE Account 56647 **op1** external "" false

2014-04-15T01:55:48.574 admin "" MODEL 4654e06c-5c1f-4955-bf82-e317dcbf38f3 CREATE Account 56647 **op1** name "" op1

2014-04-15T01:55:48.574 admin "" MODEL 4654e06c-5c1f-4955-bf82-e317dcbf38f3 CREATE Account 56647 **op1** password "" *****

- Example audit log entry generated when a User Account was assigned to a User Group

The following is an example log entry that was generated when the user **op1** was assigned to the **NNMi Level 1 Operator** User Group

2014-04-15T01:55:48.597 admin "" MODEL 4654e06c-5c1f-4955-bf82-e317dcbf38f3 CREATE UserGroupMember 56650 5486f4cf-a3e0-4f24-abd6-28f5169f9f92 account "" 56647/**op1**

2014-04-15T01:55:48.597 admin "" MODEL 4654e06c-5c1f-4955-bf82-e317dcbf38f3 CREATE UserGroupMember 56650 5486f4cf-a3e0-4f24-abd6-28f5169f9f92 **userGroup** "" 141/**level1**

- Example audit log entry generated when a User Account password was changed:

The following is an example log entry that was generated when the **op2** User Account password was changed:

Note: The first user name is the name of the user making the change. The second user name is the account name for which the password is changed.

2014-04-15T02:04:39.121 **admin** "" MODEL 0ae97c60-3035-46e0-a20c-20b6da04615f UPDATE Account 56645 **op2 password** ***** *****

Chapter 4: Resilience

HPE Network Node Manager i Software (NNMi) supports two different approaches to protecting the NNMi data in case of hardware failure:

- NNMi application failover provides for disaster recovery by maintaining a copy of the embedded NNMi database transaction logs on an identically configured system. (If NNMi uses an Oracle database, the two systems connect to the same database at different times.)
- Running NNMi in a high availability (HA) cluster provides for nearly one hundred percent availability of the NNMi management server by maintaining the embedded NNMi database and configuration files on a shared disk. (If NNMi uses an Oracle database, the shared disk contains the NNMi configuration files, and the two systems connect to the same database at different times.)

In both approaches, if the current NNMi management server fails, the second system automatically becomes the NNMi management server.

The following table compares several aspects of these two approaches to NNMi data resilience.

Note: If you have purchased NNMi Premium or NNMi Ultimate, you need to use the license keys you requested from the HPE Password Delivery Center for use with application failover or high availability. Be sure to request the following:

- High Availability: Obtain a license key for the virtual IP address of the NNMi HA resource group. This license key is initially used on the primary server and then used on the secondary server when needed.
- Application Failover: Obtain two license keys; one for the physical IP address of the primary server and one for the physical IP address of the standby server.

NNMi Data Resilience Comparison

Item for Comparison	NNMi Application Failover	NNMi Running in an HA Cluster
Required software products	NNMi or NNMi Advanced	<ul style="list-style-type: none"> • NNMi or NNMi Advanced • A separately purchased HA product
Time to fail over	Under normal conditions, 5-30 minutes depending on the number of NNM iSPIs installed.	Under normal conditions, 5-30 minutes depending on the number of NNM iSPIs installed.
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 connect to the NNMi console using the new IP address. Some applications follow the movement of the NNMi management server, but most (including the NNM iSPIs) do not.	Complete. All connections use the virtual IP address of the HA cluster, which does not change on failover.

NNMi Data Resilience Comparison, continued

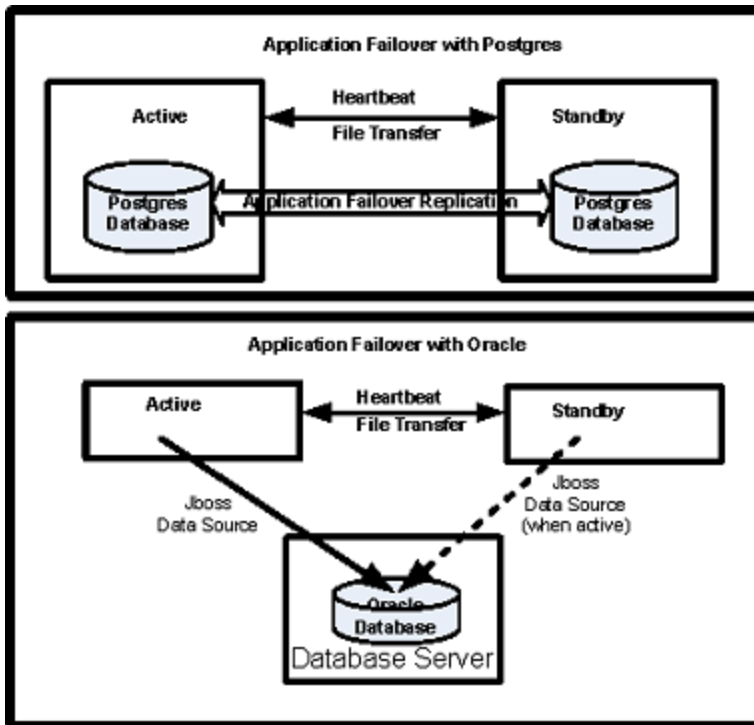
Item for Comparison	NNMi Application Failover	NNMi Running in an HA Cluster
Relative proximity of active and standby servers	LAN or WAN	LAN or WAN (some HA products only)
Licenses installed	<ul style="list-style-type: none"> • License keys on the initial active server. • License keys on the initial standby server. 	License keys on the initial active server and managed on the shared disk.
Support for NNM iSPIs	Support varies. See the documentation for each NNM iSPI.	
Interaction with Global Network Management	<ul style="list-style-type: none"> • Can configure each global manager for application failover or HA. • Can configure each regional manager for application failover or HA. • Each of these configurations requires two physical or virtual systems.^a • If a global manager or regional manager fails over, 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.

This section contains the following chapters:

- ["Configuring NNMi for Application Failover" on the next page](#)
- ["Configuring NNMi in a High Availability Cluster" on page 150](#)

^aVirtual machine support for HA is dependent on HA software vendors' support of virtual systems.

Configuring NNMi for Application Failover



Many information technology professionals depend on HPE Network Node Manager i Software (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.

This chapter contains the following topics:

- ["NNMi Application Failover Overview" on the next page](#)
- ["Application Failover Requirements" on the next page](#)
- ["Set Up NNMi for Application Failover" on page 124](#)
- ["Using the Application Failover Feature" on page 128](#)
- ["Returning to the Original Configuration Following a Failover" on page 133](#)
- ["NNM iSPi and Application Failover" on page 133](#)
- ["Integrated Applications" on page 137](#)
- ["Disabling Application Failover" on page 138](#)
- ["Administrative Tasks and Application Failover " on page 140](#)
- ["Network Latency/Bandwidth Considerations" on page 146](#)

NNMi Application Failover Overview

After configuring your systems to use the application failover feature, NNMi detects an NNMi management server failure and triggers a secondary server to assume NNMi functionality.

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

- **Active:** The server running the NNMi processes.
- **Standby:** The system in the NNMi cluster that is waiting for a failover event; this system is not running NNMi processes.
- **Cluster Member:** A process running on a system that is using JGroups technology to connect to a cluster; you can have multiple members on a single system.
- **Cluster Manager:** The `nnmcluster` process and tool used to monitor and manage the servers for the application failover feature.

Application Failover Requirements

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

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

For application failover to work successfully, the NNMi management servers must meet the following requirements:

- Both NNMi management servers must be running the same type of operating system. For example, if the active server is running a Linux operating system, the standby server must also be running a Linux operating system.
- Both NNMi management servers must be running the same NNMi version. For example, if NNMi 10.30 is running on the active server, the identical NNMi version, NNMi 10.30, must be on the standby server. The NNMi patch levels must also be the same on both servers.
- The system password must be the same on both NNMi management servers.
- Do not completely disable HTTP access to NNMi before configuring application failover. After successfully configuring the application failover cluster, you can disable HTTP and other unencrypted access.
- For NNMi installations on Windows operating systems, the `%NnmDataDir%` and `%NnmInstallDir%` system variables must be set to identical values on both servers.
- Both NNMi management servers must be running the same database. For example, both NNMi management servers must be running Oracle or both NNMi management servers must be running the embedded database. You cannot mix the two database types if you plan to use the application failover feature.
- Both NNMi management servers must have identical licensing attributes. For example, the node counts and licensed features must be identical.
- Do not enable application failover until NNMi is in an advanced stage of initial discovery. For more information see ["Evaluate Discovery" on page 67](#).

For application failover to function correctly, the active and standby servers must have unrestricted network access to each other. After meeting this condition, complete the steps shown in "[Set Up NNMi for Application Failover](#)" below. For more information see "[NNMi and NNM iSPI Default Ports](#)" on page 433.

Note: If you have purchased NNMi (only), NNMi Advanced, and the NNM iSPI NET features bundled with NNMi there are two types of licenses for use with application failover environments:

- Production - This is the main license that is purchased for NNMi, NNMi Advanced, or NNM iSPI NET whether you have an application failover or high availability environment. Associate this license with the IP address of the primary server.
- Non-production - This license is purchased separately for use in application failover environments. Associate this license with the IP address of the secondary (standby) server.

Do not use production and non-production licenses on the same server.

If you have purchased NNMi Premium or NNMi Ultimate, instead of using a non-production license as directed, you need to use the license key or license keys you requested from the HPE Password Delivery Center for use with application failover. Obtain two license keys; one for the physical IP address of the primary server and one for the physical IP address of the standby server.

Also see the documentation for each NNM iSPI, available at:
<http://h20230.www2.hp.com/selfsolve/manuals>.

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.

During an NNMi installation or upgrade, the NNMi installation chooses a network interface for NNMi Cluster communications. The network interface chosen is generally the first non-loopback interface on the system. When the NNMi Cluster is configured, the configuration uses the chosen interface. If you have to adjust the interface, do the following:

1. Edit the following file:
 - *Windows:* %NnmDataDir%\conf\nnm\props\nms-cluster-local.properties
 - *Linux:* \$NnmDataDir/conf/nnm/props/nms-cluster-local.properties
2. Adjust the `com.hp.ov.nms.cluster.interface` parameter to point to the desired interface.

Set Up NNMi for Application Failover

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

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

For application failover to work successfully, the NNMi management servers must meet the following requirements:

- Both NNMi management servers must be running the same type of operating system. For example, if the active server is running a Linux operating system, the standby server must also be running a Linux operating system.

- Both NNMi management servers must be running the same NNMi version. For example, if NNMi 10.30 is running on the active server, the identical NNMi version, NNMi 10.30, must be on the standby server. The NNMi patch levels must also be the same on both servers.
- The system password must be the same on both NNMi management servers.
- Do not completely disable HTTP access to NNMi before configuring application failover. After successfully configuring the application failover cluster, you can disable HTTP and other unencrypted access.
- For NNMi installations on Windows operating systems, the %NmDataDir% and %NmInstallDir% system variables must be set to identical values on both servers.
- Both NNMi management servers must be running the same database. For example, both NNMi management servers must be running Oracle or both NNMi management servers must be running the embedded database. You cannot mix the two database types if you plan to use the application failover feature.
- Both NNMi management servers must have identical licensing attributes. For example, the node counts and licensed features must be identical.
- Do not enable application failover until NNMi is in an advanced stage of initial discovery. For more information see ["Evaluate Discovery" on page 67](#).

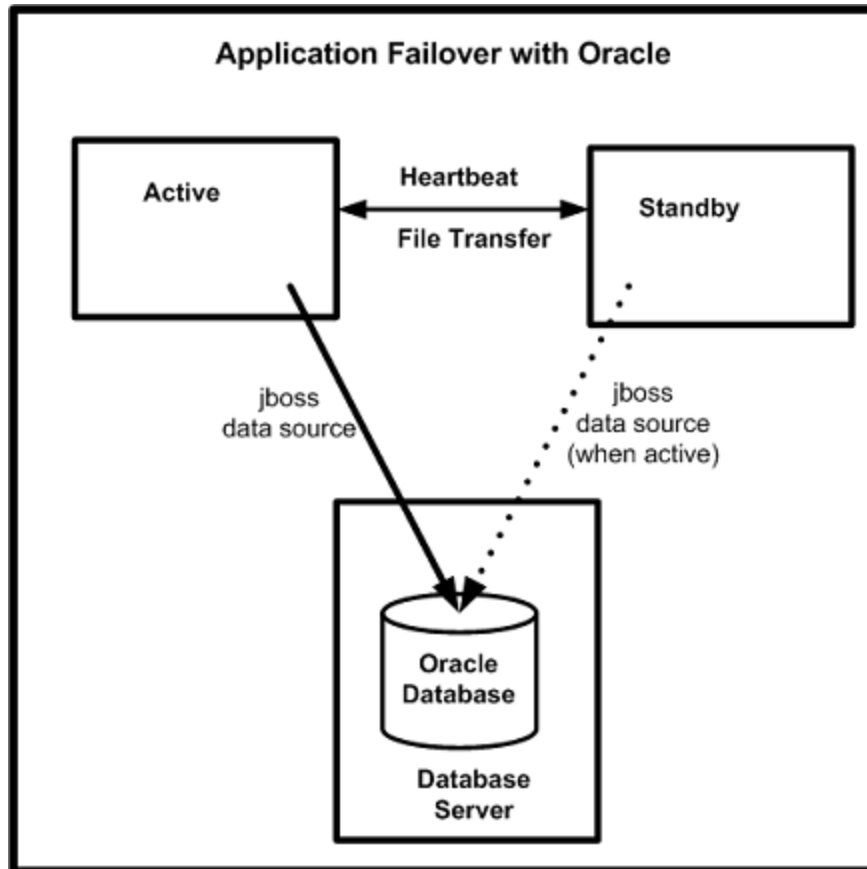
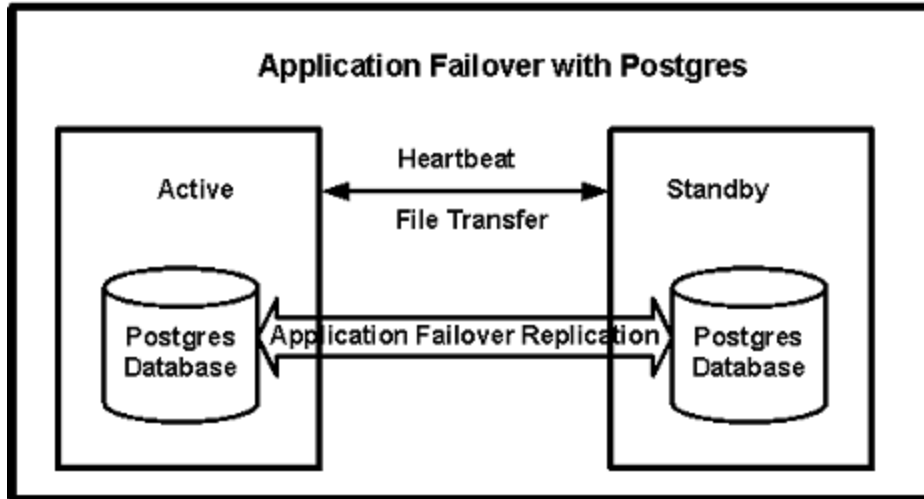
For application failover to function correctly, the active and standby servers must have unrestricted network access to each other. After meeting this condition, complete the steps shown in ["Set Up NNMi for Application Failover" on the previous page](#). For more information see ["NNMi and NNM iSPI Default Ports" on page 433](#).

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.

During an NNMi installation or upgrade, the NNMi installation chooses a network interface for NNMi Cluster communications. The network interface chosen is generally the first non-loopback interface on the system. When the NNMi Cluster is configured, the configuration uses the chosen interface. If you have to adjust the interface, do the following:

1. Edit the following file:
 - *Windows:* %NmDataDir%\conf\nnm\props\nms-cluster-local.properties
 - *Linux:* \$NmDataDir/conf/nnm/props/nms-cluster-local.properties
2. Adjust the `com.hp.ov.nms.cluster.interface` parameter to point to the desired interface.
1. Install NNMi on the active server, server X, and the standby server, server Y, as described in the HPE Network Node Manager i Software Interactive Installation Guide as shown in the following diagram:

Setting up Application Failover in NNMi



2. For each license on server X, obtain the required license for server Y and install it onto server Y as described in ["Apply Licenses" on page 241](#).
3. Run the `ovstop` command on each server to shut down NNMi.

Note: If you are using application failover with Oracle as your database, your NNMI processes on the standby server should already be stopped.

4. If you are using application failover with Oracle as your database, follow the configuration steps in ["Manually Configuring NNMI for Application Failover" on page 427](#).

Configuring your Cluster with the NNMI Cluster Setup Wizard (Embedded Database Users only)

The NNMI Cluster Setup Wizard automates the process of configuring a cluster within NNMI for use with Application Failover. The wizard lets you:

- Specify and validate cluster nodes
- Define cluster properties and ports
- Merge the `nmm-key.p12` and `nmm-trust.p12` file content for both nodes into a single `nmm-key.p12` and `nmm-trust.p12` file

1. Launch the Cluster Setup Wizard by entering the following into a supported Web browser:

```
http://<NNMIserv>:<port>/cluster
```

- `<NNMIserv>` 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**.
 4. On the Communication Results page, review the communication verification results. If an error occurs, click **Previous** and fix the problem; otherwise, click **Next**.
A green status message indicates the connection to the remote cluster node is successful.
 5. On the Define Cluster Properties page, enter the **Cluster Name**, define the **Backup Interval** (in hours), and specify whether to enable automatic failover. Click **Next**.
 6. On the Define Cluster Ports page, enter **Starting Cluster Port** and **File Transfer Port** values.

Note: The NNMI Cluster uses 4 contiguous ports beginning with the **Starting Cluster Port**.

7. Click **Next**.
8. Review the summary information provided. 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 successfully written to the configuration files.
9. Immediately stop NNMI on both nodes by running the `ovstop` command on both nodes.
10. Verify the two nodes are able to cluster by running the `nmmcluster` command on both nodes. If the nodes are not able to cluster, then see ["Manually Configuring NNMI for Application Failover" on page 427](#).
11. Start NNMI on the desired active node using the `nmmcluster` command. Wait for NNMI to report ACTIVE (see ["Manually Configuring NNMI for Application Failover" on page 427](#)).
12. Start the standby node using the `ovstart` command.

Setting Cluster Communications (Optional)

During installation, NNMi queries all Network Interface Cards (NICs) on the system to find one to use for cluster communications (the first available NIC is chosen). If your system has multiple NICs, you can choose which NIC to use for `nmcluster` operations by doing the following:

1. Run `nmcluster -interfaces` to list all available interfaces. For more information, see the *nmcluster* reference page, or the UNIX manpage.

2. Edit the following file:

- Windows:

```
%NnmDataDir%\conf\nnm\props\nms-cluster-local.properties
```

- Linux:

```
$NnmDataDir/conf/nnm/props/nms-cluster-local.properties
```

3. Look for a line containing text similar to the following:

```
com.hp.ov.nms.cluster.interface =<value>
```

4. Change the value as desired.

Note: The interface value must pertain to a valid interface; otherwise, the cluster might not be able to start.

5. Save the `nms-cluster-local.properties` file.

Note: The `com.hp.ov.nms.cluster.interface` parameter permits NNMi administrators to select the communication interface used for `nmcluster` communication. This interface is not the interface used for the embedded database or Secure Sockets Layer communication.

Note: To configure communications so that application failover is honored by a specific interface, use the IP address in the `com.hp.ov.nms.cluster.member.hostnames` parameter, as opposed to using a hostname. Set the `com.hp.ov.nms.cluster.member.hostnames` parameter in the following file:

Windows:

```
%NnmDataDir%\shared\nnm\conf\props\nms-cluster.properties
```

Linux:

```
$NnmDataDir/shared/nnm/conf/props/nms-cluster.properties
```

Using the Application Failover Feature

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

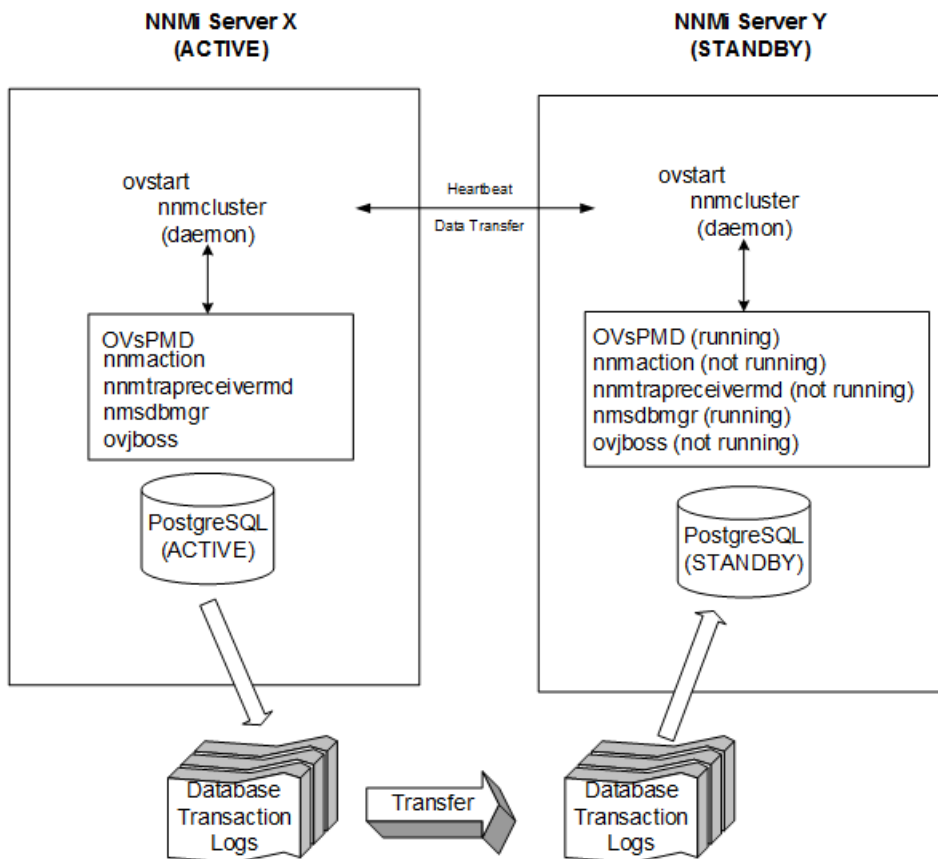
- **daemon mode:** The cluster manager process runs in the background, and uses the **ovstop** and **ovstart** commands to start and stop the NNMi services.
- **interactive mode:** The cluster manager runs an interactive session in which the NNMi administrator can view and change cluster attributes.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

For more information, see the *nmcluster* reference page, or the Linux manpage.

Application Failover Behavior Using the Embedded Database

The following diagram shows the application failover configuration for two NNMi management servers using the embedded database. Refer to this diagram while reading the rest of this chapter.

Application Failover Configuration (embedded database)



Note: If you remove a standby server from a cluster, run that server as a standalone server, and then add it back into the cluster, you might receive a database error. If this occurs, run the following command from the command line: `nmcluster dbsync`.

NNMi 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 on the standby server on 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 only sent from one node to another if and when needed, and given the regular transmission of database transaction files, the need for sending large database backup files should be infrequent.

Note: For both the active and standby nodes, if you have a firewall enabled, ensure that the port you are using for the embedded database (port 5432 by default) is open. This port is set in the following file:

Windows: %NNM_CONF%\nmm\props\nms-local.properties

Linux: \$NNM_CONF/nmm/props/nms-local.properties

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

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

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

If the active node becomes unavailable for any reason, the standby node becomes active by running an `ovstart` command to start the NNMI services. The standby NNMI management server imports the transaction logs before starting the remaining NNMI services.

If the active NNMI system fails, the standby system begins discovery and polling activities. This transition keeps NNMI monitoring and polling your network while you diagnose and repair the failed system.

Note:

- NNMI automatically resynchronizes topology, state, and status following an application failover.
- Avoid stopping NNMI during the resynchronization.

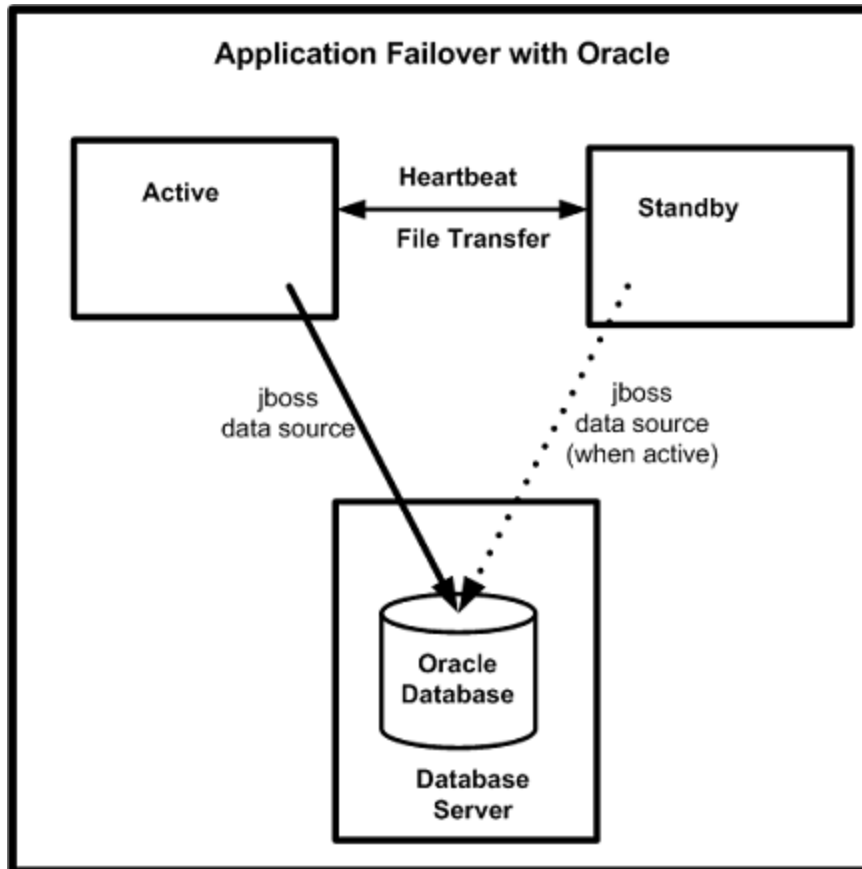
To help ensure resynchronization has completed, NNMI should remain running for several hours following the application failover. The actual time required depends on the number of nodes and the volume of state changes and trap data received while performing the resynchronization.
- If NNMI must be stopped before the resynchronization is finished, the resynchronization should be run again and allowed to complete.
- To perform a manual resynchronization of the entire management server, run:


```
nnmmoderediscover.ovpl -all -fullsync
```

Application Failover Behavior Using an Oracle Database

The following diagram shows the application failover configuration for two NNMI management servers using an Oracle database. Refer to this diagram while reading the rest of this chapter.

Application Failover Configuration (Oracle database)



If the active node becomes unavailable for any reason, the standby node becomes active by running an **ovstart** command to start the NNMi services.

If the active NNMi system fails, the standby system begins discovery and polling activities. This transition keeps NNMi monitoring and polling your network while you diagnose and repair the failed system.

Note:

- Updates to Status and Incidents could be delayed as NNMi resynchronizes following an application failover.
- If you see the following message during this resynchronization, it does not indicate a problem:
The Causal Engine's large queue size is causing delayed updates to Status and Incidents. This could be due to resynchronization following an upgrade, application failover, restore from backup, or a manual resynchronization.
- Do not stop NNMi during this resynchronization. To ensure resynchronization has completed, keep NNMi running for several hours following the application failover.

Application Failover Scenarios

Several possible problems can cause the active NNMi management server to stop sending heartbeats, and to initiate a failover:

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

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

Additional ovstart and ovstop Options

When you use the **ovstop** and **ovstart** commands on NNMi management servers configured for application failover, NNMi runs the following commands:

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

Note: If you run an **ovstop** command, NNMi does not failover to the standby node. HPE designed the **ovstop** command to support temporary maintenance stoppages. To manually initiate a failover, use the **-failover** option with the **ovstop** command. For more information, see the *ovstop* reference page, or the Linux manpage.

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 failover mode and then stops the local daemon-mode cluster process. No failover occurs. This command is equivalent to: **nnmcluster -disable -shutdown**
- **ovstop -cluster:** This command stops both the active and standby nodes, removing them both from the cluster. This command is equivalent to: **nnmcluster -halt**

Note: If you run the **shutdown** command on NNMi management servers running Linux operating systems, the **ovstop** command runs automatically and disables application failover. That might not be your desired result. To control application failover during maintenance windows, use the **nnmcluster -acquire** and **nnmcluster -relinquish** commands to set the active and standby nodes the way you want them before running the shutdown command. For more information see the *nnmcluster* reference page, or the Linux manpage.

Application Failover Incidents

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

- *NnmClusterStartup:* The NNMi cluster was started, and no active node was present. Therefore the node was started in the active state. This incident has a Normal severity.
- *NnmClusterFailover:* The NNMi cluster detected a failure of the active node. The standby node was then enabled and NNMi services started on the new active node. This incident has a Major severity.

Returning to the Original Configuration Following a Failover

If the active node fails and the standby node is functioning as the active node, after the former active node is fixed, you can return to the original configuration.

Perform the following steps:

1. Fix the problem with the former active node.
2. Run the following command on the desired active node to return to the original configuration:

```
nnmcluster -acquire
```

For more information, see the *nnmcluster* reference page, or the Linux manpage.

NNM iSPIs and Application Failover

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

- The NNM iSPI runs on the NNMi management server.
- *Embedded database only.* The NNM iSPI uses the same embedded database instance as NNMi.
- *Oracle database only.* The NNM iSPI must use a unique Oracle database instance from that used by NNMi.

The NNM iSPI Performance for Metrics and the NNM iSPI Performance for Traffic are exceptions to this description. If you plan to configure the NNMi application failover feature, you must install these iSPIs on dedicated servers. In this case, the iSPIs automatically connect to the new NNMi management server after failover occurs. See the NNM iSPI Performance for Metrics and NNM iSPI Performance for Traffic documentation for more information on installing these iSPIs when NNMi is installed in an application failover environment.

For more information, see *Support for Application Failover* in the NNM iSPI Performance for Metrics, the NNM iSPI Performance for QA, or the NNM iSPI Performance for Traffic help.

NNM iSPI Installation Information for Embedded Database

Deploying NNM iSPI in an Existing Application Failover Environment - Embedded Database

To install or upgrade an NNM iSPI (IP Telephony, MPLS, or Multicast) on an NNMi management server that is already part of an application failover cluster, follow these steps:

1. (for NNM iSPI upgrade only) As a precaution, back up all necessary configuration details and NNMi data:
 - Run the `nnmconfigexport.ovp1` script on both the active and standby NNMi management servers before proceeding.
 - Back up the NNMi data on both the active and standby NNMi management servers before proceeding. For information, see ["Backup Scope" on page 223](#).
2. On the active NNMi management server run the `nnmcluster -dbsync` command and wait for the command to complete.
3. Disable application failover by following these steps:

- a. On the standby NNMi management server, run the following command:


```
nmcluster -shutdown
```
 - b. 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
 - c. Comment out the `com.hp.ov.nms.cluster.name` option and save the file.
 - d. 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
 - e. Remove the following lines, which are automatically added by application failover. This is an example of what these lines could look like. These lines might look slightly different on your server.


```
# The following lines were added by the NNM cluster.

archive_command = ...
archive_timeout = 900
max_wal_senders = 4
archive_mode = 'on'
wal_level = 'hot_standby'
hot_standby = 'on'
wal_keep_segments = 500
listen_addresses = 'localhost,16.78.61.68'
```

Make sure to save your changes.
 - f. Remove the files `recovery.conf` and `recovery.done` if they exist at the following location:
 - *Windows*: %NnmDataDir%\shared\nnm\databases\Postgres\
 - *Linux*: \$NnmDataDir/shared/nnm/databases/Postgres/
 - g. Create the following trigger file, which tells Postgres to stop running in standby mode and to start fully running:
 - *Windows*: %NnmDataDir%\tmp\postgresTriggerFile
 - *Linux*: %NnmDataDir%/tmp/postgresTriggerFile\
4. Install or upgrade the NNM iSPI on both active and standby servers and enable application failover by following these steps:
 - a. Run the `ovstart` command on the standby NNMi management server. This brings up NNMi services in the standalone (unclustered) state.
 - b. Install or upgrade the NNM iSPI on the standby NNMi management server as described in the iSPI installation or upgrade document.
 - c. Run the `nmcluster -halt` command on the active NNMi management server.
 - d. Edit the following file on the active NNMi management server:
 - *Windows*: %NnmDataDir%\shared\nnm\conf\props\nms-cluster.properties
 - *Linux*: \$NnmDataDir/shared/nnm/conf/props/nms-cluster.properties

- e. Comment out the `com.hp.ov.nms.cluster.name` option and save the file.
 - f. Run the `ovstart` command on the active NNMi management server. This brings up NNMi services in the standalone (unclustered) state.
 - g. Install or upgrade the NNM iSPI on the active NNMi management server as described in the iSPI installation or upgrade document.
 - h. Edit the following file on **both** the active and standby NNMi management servers:
 - *Windows:* `%NnmDataDir%\shared\nnm\conf\props\nms-cluster.properties`
 - *Linux:* `$NnmDataDir/shared/nnm/conf/props/nms-cluster.properties`
 - i. Uncomment the `com.hp.ov.nms.cluster.name` option and save each file.
 - j. Run the `nmcluster -daemon` command on the active NNMi management server.
 - k. Wait a few minutes for the active NNMi management server to become the first active node in the cluster. Run the `nmcluster -display` command on the active NNMi management server and search the displayed results for the term `ACTIVE` as in `ACTIVE_NNM_STARTING` or `ACTIVE_SomeOtherState`. Do not continue until you know that the active NNMi management server is the active node.
5. Run the `nmcluster -daemon` command on the standby NNMi management server.
 6. On the active node, run the following command:


```
nmcluster -dbsync
```

Deploying NNM iSPI with NNMi and then Configuring Application Failover - Embedded Database

To install an NNM iSPI (IP Telephony, MPLS, or Multicast) with NNMi and then configure application failover, follow these steps:

1. Install NNMi and NNM iSPI on the primary and secondary servers.
2. Install the non production licenses for the NNM iSPI on both the servers.
3. Configure application failover as described in ["Set Up NNMi for Application Failover" on page 124](#). The NNM iSPI is automatically configured for application failover.

NNM iSPI Installation Information for Oracle Database

Deploying NNM iSPIs in an Existing NNMi Application Failover Environment - Oracle Database

To install or upgrade an NNM iSPI (IP Telephony, MPLS, or Multicast) on an NNMi management server that is already part of an application failover cluster that uses Oracle database, follow these steps:

1. (for NNM iSPI upgrade only) As a precaution, back up all necessary configuration details and NNMi data:
 - Run the `nmconfigexport.ovpl` script on both the active and standby NNMi management servers before proceeding.
 - Back up the NNMi data on both the active and standby NNMi management servers before proceeding. For information, see ["Backup Scope" on page 223](#).
 - Back up all NNMi data on the Oracle database using appropriate database commands.

2. Disable the application failover by following these steps:
 - a. Run the `nmcluster -shutdown` command on the active NNMi management server.
 - b. Wait a few minutes for the old standby NNMi Management Server to become the new active NNMi Management Server.
 - c. Run the `nmcluster -display` command on the new active (old standby) NNMi Management Server.
 - d. Search the displayed results for the `ACTIVE_NNM_RUNNING` status. Repeat the previous step until you see the `ACTIVE_NNM_RUNNING` status.
 - e. Run the `nmcluster -halt` command on the new active (old standby) NNMi Management Server.
 - f. Run the `nmcluster -display` command repeatedly on the new active (old standby) until you no longer see the `DAEMON` process.
 - g. Run the `ovstatus -a` command on both the servers and ensure that NNMi processes are all stopped.
 - h. On both active and standby NNMi Management Servers, edit the following file and comment out the `com.hp.ov.nms.cluster.name` option.
 - *Windows:* `%NnmDataDir%\shared\nnm\conf\props\nms-cluster.properties`
 - *Linux:* `$NnmDataDir/shared/nnm/conf/props/nms-cluster.properties`
3. Install or upgrade an NNM iSPI first on the active and then on the standby NNMi Management Server by following these steps:
 - a. On the active NNMi Management Server, run the `ovstart` command. This brings up NNMi services in the standalone (unclustered) state.
 - b. Install or upgrade the NNM iSPI on the NNMi management server as described in the iSPI installation or upgrade documents.
 - c. Run the `ovstop` command on the active NNMi Management Server.
 - d. On the standby NNMi Management Server, run the `ovstart` command. This brings up NNMi services in the standalone (unclustered) state.
 - e. Install or upgrade the NNM iSPI on the NNMi management server as described in the iSPI installation or upgrade documents.
 - f. Run the `ovstop` command on the standby NNMi Management Server.
4. Enable application failover by following these steps:
 - a. On both the active and the standby NNMi Management Servers, edit the following file and uncomment the `com.hp.ov.nms.cluster.name` option.
 - *Windows:* `%NnmDataDir%\shared\nnm\conf\props\nms-cluster.properties`
 - *Linux:* `$NnmDataDir/shared/nnm/conf/props/nms-cluster.properties`
 - b. Run the `nmcluster -daemon` command on the active NNMi management server.
 - c. Wait a few minutes for the active NNMi management server to become the first active node in the cluster. Run the `nmcluster -display` command on the active NNMi management server and search the displayed results for the term `ACTIVE` as in `ACTIVE_NNM_STARTING` or `ACTIVE_SomeOtherState`. Do not continue with the next steps until you know that the active NNMi management server is the active node.
 - d. Run the `nmcluster -daemon` command on the standby NNMi management server.

Deploying NNM iSPIs with NNMi and then Configuring Application Failover - Oracle Database

To install an NNM iSPI (IP Telephony, MPLS, or Multicast) with NNMi and then configure application failover, follow these steps:

1. Install NNMi and the required NNM iSPIs (IP Telephony, Multicast, or MPLS) in server 1 (primary NNMi Management Server) with Oracle as a primary server.
2. Run the `ovstop` command on the primary NNMi Management Server.
3. Install NNMi in server 2 (secondary NNMi Management Server) with Oracle as a secondary server.
4. Merge the keystores on one server and copy the keystores to both the primary and the secondary servers. For more information, see ["Working with Certificates in Application Failover Environments" on page 257](#).
5. Start NNMi services on the secondary NNMi Management Server.
6. Install the required NNM iSPI in the secondary NNMi Management Server with Oracle as a secondary server.

Note: Ensure that you enter the same web service client details used when installing NNM iSPIs on the primary NNMi Management Server.

7. Configure application failover as described in ["Set Up NNMi for Application Failover" on page 124](#).

Note: Keep in mind that you have already merged keystores as part of this procedure. Skip those steps while configuring application failover.

Integrated Applications

When other HPE Software or third-party products are integrated with NNMi, the affect of NNMi application failover on an integration depends on how a product communicates with NNMi. For more information, see the appropriate integration document.

If an integrated product must be configured with information about the NNMi management server, the following information applies:

- If long-term, you can update the NNMi management server information within the integrating product configuration. For more information, see the appropriate integration document.
- If the outage appears to be temporary, you can resume using the integrating product after server X returns to service. To return server X to service, follow these steps:
 1. On server X, run the following command:


```
nnmcluster -daemon
```

 Server X joins the cluster and assumes a standby state.
 2. On server X, run the following command:


```
nnmcluster -acquire
```

 Server X changes to the active state.

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

Disabling Application Failover

The following information explains how to completely disable application failover. Complete the following instructions, including actions on both the active and standby NNMi management servers configured in the application failover cluster.

Note: If you have purchased NNMi (only), NNMi Advanced, and the NNM iSPI NET features bundled with NNMi there are two types of licenses for use with application failover:

- Production - This is the main license that is purchased for NNMi, NNMi Advanced, or NNM iSPI NET whether you have an application failover or high availability environment. Associate this license with the IP address of the primary server.
- Non-production - This license is purchased separately for use in application failover environments. Associate this license with the IP address of the secondary (standby) server.

If you have purchased NNMi Premium or NNMi Ultimate, instead of using a non-production license as directed, you need to use the license key or license keys you requested from the HP Password Delivery Center for use with application failover. Obtain two license keys; one for the physical IP address of the primary server and one for the physical IP address of the standby server.

Caution: Do not use production and non-production licenses on the same server.

Also see the documentation for each NNM iSPI, available at:
<http://h20230.www2.hp.com/selfsolve/manuals>.

1. Run `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) until you no longer see a DAEMON process.
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 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
11. Remove the following lines, which are automatically added by application failover. This is an example of what these lines could look like. These lines might look slightly different on your server.

```
# The following lines were added by the NNM cluster.
archive_command = ...
archive_timeout = 900
max_wal_senders = 4
archive_mode = 'on'
wal_level = 'hot_standby'
hot_standby = 'on'
wal_keep_segments = 500
listen_addresses = 'localhost,16.78.61.68'
```

Make sure to save your changes.

12. If these are Windows NNMi management servers, navigate to the Services (Local) console and do the following on each server:
 - a. Set the Startup type for the HPE NNM Cluster Manager to Disabled.
 - b. Set the Startup type for the HP OpenView Process Manager to Automatic.
13. Create the following trigger file, which tells Postgres to stop running in standby mode and to start fully running:


```
Windows: %NnmDataDir%\tmp\postgresTriggerFile
Linux: $NnmDataDir/tmp/postgresTriggerFile
```
14. Run the **ovstart** command on the former active NNMi management server only. In the application failover configuration, this is the NNMi management server that has a permanent NNMi license.
15. If you were using a non-production license on the former standby server. Do not run the **ovstart** command on the former standby NNMi management server. In the application failover configuration, this is the NNMi management server that has a non-production license. To run this NNMi management server as a standalone server, you must purchase and install a permanent license. For more information, see ["Apply Licenses" on page 241](#).
16. If both NNMi management servers start successfully, then 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 `nms-cluster.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 `nms-cluster.properties` file, then remove the directory that equates to the new value.

17. 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

Administrative Tasks and Application Failover

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

Restoring NNMi Failover Environment

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 ["Backing up NNMi Data" on page 223](#).
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.
 - **Embedded Database:** Use `nmmrestore.ovpl` command to do a full restore. For more information, see ["Backup and Restore Strategies" on page 228](#).
 - **Oracle Database:** Use the restore command similar to the following to restore only the system files. For more information, see ["Restore File System Files Only" on page 229](#).

```
nmmrestore.ovpl -partial -source nmi_backups\offline\

```

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: `%NmDataDir%\shared\nm\conf\props\nms-cluster.properties`
 - Linux: `$NmDataDir/shared/nm/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 ["Managing Certificates" on page 243](#).

Application Failover and NNMi Patches

Both NNMi management servers must be running the same NNMi version and patch level. To add patches to the active and standby NNMi management servers, use one of the following procedures:

- ["Applying Patches for Application Failover \(Shut Down Both Active and Standby\)" on the next page](#)
Use this procedure when you are not concerned with an interruption in network monitoring.
- ["Applying Patches for Application Failover \(Keep One Active NNMi Management Server\)" on page 142](#)
Use this procedure when must avoid any interruptions in network monitoring.

Applying Patches for Application Failover (Shut Down Both Active and Standby)

This procedure results in both NNMI management servers being non-active for some period of time during the patch process. To apply patches to 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 standby NNMI management servers before proceeding.
2. As a precaution, back up your NNMI data on both the active and standby NNMI management servers before proceeding. For information, see ["Backup Scope" on page 223](#).
3. Note the `com.hp.ov.nms.cluster.name` property value in the `nms-cluster.properties` file. You will need this value after the patch installation. This file is in the following location:
 - Windows: `%nnmDataDir%\shared\nnm\conf\props\nms-cluster.properties`
 - Linux: `$nnmDataDir/shared/nnm/conf/props/nms-cluster.properties`
4. As a precaution, on the active NNMI management server, do the following steps:
 - a. Run the `nnmcluster` command.
 - b. Embedded database only: After NNMI prompts you, type `dbsync`, then press Enter. Review the displayed information to make sure it includes the following messages:

ACTIVE_DB_BACKUP: This means that the active NNMI management server is performing a new backup.

ACTIVE_NNM_RUNNING: This means that the active NNMI management server completed the backup referred to by the previous message.

STANDBY_READY: This shows the previous status of the standby NNMI management server.

STANDBY_RECV_DBZIP: This means that the standby NNMI management server is receiving a new backup from the active NNMI management server.

STANDBY_READY: This means that the standby NNMI management server is ready to perform if the active NNMI management server fails.
5. Run the `nnmcluster-halt` command on the active NNMI management server. This shuts down all `nnmcluster` processes on both the active and standby NNMI management servers.
6. To verify there are no `nnmcluster` nodes running on either server, *complete the following steps on both the active and standby NNMI management servers*.
 - a. Run the `nnmcluster` command.
 - 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](#).
7. 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 changes.
8. Apply the NNMI patch to the active NNMI management server using the instructions provided with the patch.

9. On the active NNMI management server, uncomment the `com.hp.ov.nms.cluster.name` parameter in the `nms-cluster.properties` file.

Note: During patch installation the `com.hp.ov.nms.cluster.name` property value is replaced with the NNMI default value. After you uncomment the line that contains the `com.hp.ov.nms.cluster.name` parameter, you also need to replace the `com.hp.ov.nms.cluster.name` property value with the value that was configured before the patch was installed.

- a. Edit the following file:
 - o *Windows:* `%NNM_SHARED_CONF%\props\nms-cluster.properties`
 - o *Linux:* `$NNM_SHARED_CONF/props/nms-cluster.properties`
 - b. Uncomment the `com.hp.ov.nms.cluster.name` parameter in the `nms-cluster.properties` file on the active NNMI management server.
 - c. Replace the default value of the `com.hp.ov.nms.cluster.name` property with the name that was configured in `nms-cluster.properties` before the patch was installed.
 - d. Save your changes.
10. Run the `ovstart` command on the active NNMI management server.
 11. Verify that the patch 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.
 12. Run the `nmcluster -dbsync` command to create a new backup.
 13. 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 [step a](#) through [step c](#).
 14. Apply the NNMI patch to the standby NNMI management server.
 15. On the standby NNMI management server, uncomment and update the `com.hp.ov.nms.cluster.name` parameter in the `nms-cluster.properties` file as shown in [step a](#) through [step d](#).
 16. Run the `ovstart` command on the standby NNMI management server.
 17. If you installed the NNM iSPI Performance for QA, the NNM iSPI Performance for Metrics, or the NNM iSPI Performance for Traffic; are using the application failover feature; and completed the patch process shown above, run the NNM iSPI enablement script for each NNM iSPI on both the active and standby NNMI management servers.

Applying Patches for Application Failover (Keep One Active NNMI Management Server)

This procedure results in one NNMI management server always being active during the patch process.

Note: This process results in continuous monitoring of the network, however NNMI loses the transaction logs occurring during this patch process.

To apply NNMI patches to the NNMI management servers configured for application failover, follow these steps:

1. As a precaution, run the `nmconfigexport.ovpl` script on both the active and standby NNMI management servers before proceeding.

2. As a precaution, back up your NNMI data on both the active and standby NNMI management servers before proceeding. For information, see ["Backup Scope" on page 223](#).
3. Note the `com.hp.ov.nms.cluster.name` property value in the `nms-cluster.properties` file. You will need this value after the patch installation. This file is in the following location:

Windows: %nnmDataDir%\shared\nnm\conf\props\nms-cluster.properties

Linux: \$nnmDataDir/shared/nnm/conf/props/nms-cluster.properties

4. Run `nnmcluster` on one of the nodes.
5. Enter `dbsync` on the NNMI management server used in the previous step to synchronize the two databases.

Note: The `dbsync` option works on an NNMI management server using the embedded database. Do not use the `dbsync` option on an NNMI management server configured to use an Oracle database.

6. Wait until the active NNMI management server reverts to `ACTIVE_NNM_RUNNING` and the standby NNMI management server reverts to `STANDBY_READY`. before continuing.
7. Exit or quit from the `nnmcluster` command.
8. Stop the cluster on the standby NNMI management server by running the following command on the standby NNMI management server:
nnmcluster -shutdown
9. Make sure the following processes and services terminate before continuing:
 - postgres
 - ovjboss
10. Make sure the `nnmcluster` process terminates before continuing. If the `nnmcluster` process will not terminate, manually kill the `nnmcluster` process only as a last resort.
11. 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
12. Comment out the cluster name by placing a `#` at the front of the line, then save your changes:
#com.hp.ov.nms.cluster.name = NNMIcluster
13. Install the NNMI patch on the standby NNMI management server.
14. At this point, the standby NNMI management server is patched but stopped, and the active NNMI management server is unpatched but running. Stop the active NNMI management server and immediately bring the standby NNMI management server online to monitor your network.
15. Shut down the cluster on the active NNMI management server by running the following command on the active NNMI management server:
nnmcluster -halt
16. Make sure the `nnmcluster` process terminates. If it does not terminate within a few minutes, manually kill the `nnmcluster` process.
17. On the standby NNMI management server, uncomment the cluster name from the `nms-cluster.properties` file.

Note: During patch installation the `com.hp.ov.nms.cluster.name` property value is replaced with the NNMi default value. After you uncomment the line that contains the `com.hp.ov.nms.cluster.name` parameter, you also need to replace the `com.hp.ov.nms.cluster.name` property value with the value that was configured before the patch was installed.

- 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 in the `nms-cluster.properties` file on the active NNMi management server.
 - c. Replace the default value of the `com.hp.ov.nms.cluster.name` property with the name that was configured in `nms-cluster.properties` before the patch was installed.
 - d. Save your changes.
18. Start the cluster on the standby NNMi management server by running the following command on the standby NNMi management server:
nnmcluster -daemon
 19. Install the NNMi patch on the active NNMi management server.
 20. At this point, the previous active NNMi management server is patched but offline. Bring it back into the cluster (as the standby NNMi management server) by performing the following:
 - a. Uncomment the `com.hp.ov.nms.cluster.name` parameter in the `nms-cluster.properties` file on the active NNMi management server.
 - b. Replace the default value of the `com.hp.ov.nms.cluster.name` property with the name that was configured in `nms-cluster.properties` before the patch was installed.
 - c. Start the active NNMi management server using the following command:
nnmcluster -daemon
 21. To monitor the progress, run the following command on both the active and standby NNMi management servers:
nnmcluster
Wait until the previous active NNMi management server finishes retrieving the database from the previous standby NNMi management server.
 22. After the previous active NNMi management server displays `STANDBY_READY`, run the following command on the previous active NNMi management server:
nnmcluster -acquire
 23. If you installed the NNM iSPI Performance for QA, the NNM iSPI Performance for Metrics, or the NNM iSPI Performance for Traffic; are using the application failover feature; and completed the patch process shown above, run the NNM iSPI enablement script for each NNM iSPI on both the active and standby NNMi management servers.

Application Failover and Restarting the NNMi Management Servers

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

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

1. Run the `nnmcluster -disable` command on the NNMi management server to disable the application failover feature.
2. Restart the NNMi management server.
 - a. Run the `ovstop` command on the NNMi management server.
 - b. Run the `ovstart` command on the NNMi management server.
3. Run the `nnmcluster -enable` command on the NNMi management server to enable the application failover feature.

Application Failover Control after a Communication Failure

After a communication failure between the two cluster nodes is resolved, the NNMi management server that had been running the longest before the communication failure (in other words, the previous active) is designated as the active server.

Application Failover and Recovery from a Previous Database Backup (Embedded Database Only)

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.
2. Delete or move the following directory on both the active and standby NNMi management servers:
 - *Windows:* %NnmDataDir%\shared\nnm\databases\Postgres_standby
 - *Linux:* \$NnmDataDir/shared/nnm/databases/Postgres_standby
3. Restore the database on the active NNMi management server:
 - a. Modify the following file to comment out the cluster name:
 - *Windows:* %NnmDataDir%\shared\nnm\conf\props\nms-cluster.properties
 - *Linux:* \$NnmDataDir/shared/nnm/conf/props/nms-cluster.properties
 - b. Restore the database as normal. See ["Restoring NNMi Data" on page 226](#).
 - c. Run the `ovstop` command on the active NNMi management server.
 - d. Modify the following file to uncomment the cluster name:
 - *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. 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. To verify that this step is complete, run the `nnmcluster -display` command and look for a STANDBY_READY message.

Cluster File Transfer Warning Configurations

NNMi Application Failover feature continuously synchronizes database files and configuration files by periodically transferring them from the Primary server to the Standby server. A network transport issue might

cause the file transfers to fail and thereby cause databases to be out of sync.

NNMi internally tracks the time since the last file transfer failed and generates health warnings in the NNMi Health Report when the file transfers consistently fail. Different health warnings are generated for the durations specified in the table below. These durations can be reconfigured to meet your requirements.

Durations for Generating Health Warnings

Health Warning Level	Timeout Durations
Minor	15 minutes
Major	30 minutes
Critical	45 minutes

You can reconfigure the health warning timeout durations by uncommenting and modifying the following properties in the `nms-cluster.properties` file.

- `#com.hp.ov.nms.cluster.timeout.filetransfer.MINOR = 15`
- `#com.hp.ov.nms.cluster.timeout.filetransfer.MAJOR = 30`
- `#com.hp.ov.nms.cluster.timeout.filetransfer.CRITICAL = 45`

The `nms-cluster.properties` file is located at:

Windows: `%NnmDataDir%\shared\nnm\conf\props\nms-cluster.properties`

Linux: `$NnmDataDir/shared/nnm/conf/props/nms-cluster.properties`

Note: File transfer timeout duration for minor, major, or critical should be:

- longer than the directory scan interval
- a multiple of the directory scan interval

For example, for a directory scan interval of 15 minutes, the transfer timeout durations in minutes can be 30 (15*2) for minor, 45 (15*3) for major, 60 (15*4) for critical warnings. You can confirm the directory scan interval in `nms-cluster.properties` file against the property `com.hp.ov.nms.cluster.timeout.scandir`.

See NNMi Online Help for information about accessing and using NNMi Health Report.

Network Latency/Bandwidth Considerations

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

The NNMi embedded database can become quite large, and can grow to 1GB or more even though this file is always compressed. Also, NNMi generates hundreds, or even thousands, of transaction logs during the built-in backup interval (a configuration parameter 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 an HPE test environments is shown here:

Number of nodes managed: 15,000

Number of interfaces: 100,000

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

Size of database: 850MB (compressed)

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

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

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

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

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

Application Failover and the NNMi Embedded Database

Application failover works with both the embedded and the Oracle database for NNMi 10.30. However, with Oracle, the database resides on a server that is separate from any NNMi management server. When you configure NNMi to work with an Oracle database, there is no database replication. This results in reduced network demands for application failover using an Oracle database. When using application failover with Oracle, the network uses less than 1% of the network demands as compared to using application failover with the embedded database. The information contained in this section explains NNMi traffic information related to application failover using the embedded database.

After you configure NNMi using the embedded database for application failover, NNMi does the following:

1. The active node performs a database backup, storing the data in a single ZIP file.
2. NNMi sends this ZIP file across the network to the standby node.
3. The standby node expands the ZIP file, and configures the embedded database to import transaction logs on the first startup.
4. The embedded database on the active node generates transaction logs, depending on database activity.
5. Application failover sends the transaction logs across the network to the standby node, where they accumulate on the disk.
6. When the standby node 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 complexity of the information stored within those files (some files take longer to import than other files of comparable size).
7. After the standby node imports all of the transaction logs, the database becomes available, and the standby node starts the remaining NNMi processes.
8. The original standby node is now active, and the procedure starts over at [step 1](#).

Network Traffic in and Application Failover Environment

NNMi transfers many items across the network from the active node to the standby node 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 node can verify that its files are in sync with those on the active node.
- Miscellaneous events, such as changes in parameters (enable/disable failover and others) and nodes joining or node leaving the cluster,

The first two items generate 99% of the network traffic used by application failover. This section 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 in 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 database.

Although the embedded database itself grows with database activity, it reaches a stable size for your environment, with only moderate growth over time.

Database Transaction Logs: The embedded database works by creating an empty 16 MB file, then writing database transaction information to that file. NNMi closes this file, then makes it available to application failover after 15 minutes, or after writing 16 MB 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 MB file compresses down to under 1MB. A *full* 16MB file compresses to about 8 MB. Keep in mind that during periods of higher database activity, application failover generates more transaction logs in a shorter period of time, since each file gets full faster.

An Application Failover Traffic Test

The following test resulted in an average of about 2 transaction log files per minute, with an average file size of 7 MB per file. This is due to the database activity associated with discovery of the additional 5000 nodes added with each failover event. The database in this test case eventually stabilized at about 1.1GB (as measured by the size of the backup ZIP file), with 31,000 nodes and 960,000 interfaces.

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 node became active, and the previous-active node became standby). Immediately after failover, test personnel added approximately 5,000 more nodes, waited another 4 hours to let the NNMi discovery process stabilize, then induced another failover

(failed back to the previous active node). Test personnel repeated this cycle several times with some variation in the time between failover (4 hours, then 6 hours, then 2 hours). After each failover event, test personnel measure the following:

- The size of the database backup ZIP file (created when the node first became active).
- The transaction logs: the total number of files and disk space utilization.
- The number of nodes and interfaces in the NNMI database immediately before inducing failover.
- Time to complete failover. This included the time from the initial `ovstop` command on the active node until the standby node became fully active with NNMI running.

The following table summarizes the results:

Application Failover Test Results

Hours	DB.zip Size (MB)	No. of Tx Logs	Tx Logs (GB)	Nodes	Interfaces	FailoverTime (Minutes)
4	6.5	50	.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	1100	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 node to the standby node, the transfer averaged about 5 GB every 4 hours, which is a continuous throughput of approximately 350KB/s (kilobytes per second) or 2.8 Mb/S (megabits per second).

Note: This data does not include any other application failover traffic, such as the heartbeat, file consistency checks, or other application failover communication. 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.

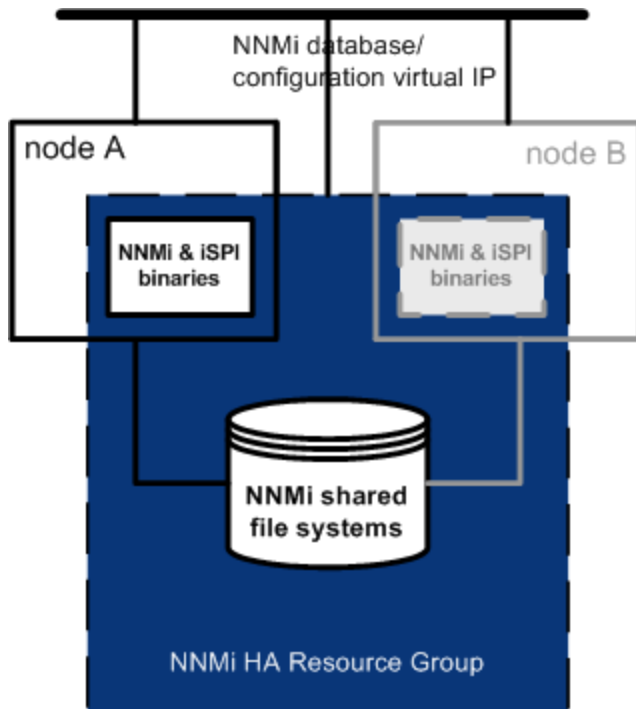
Note: The traffic generated by NNMI application failover environment is very bursty. Application failover identifies new transaction logs on the active node every five minutes and sends these logs to the standby node. Depending on network speed, the standby node should receive all of the new files in a short time, resulting in a relatively idle network for the remainder of that 5-minute interval.

Every time the active and standby nodes switch roles (the standby node becomes active and the active node becomes standby), the new active node will generate a complete database backup and send this across the network to the new standby node. This database backup also occurs periodically, backing up every 24 hours by default. Every time NNMI generates a new backup, it sends this backup to the standby node. Having this

new backup available on the standby node reduces the failover time, as all of the transaction logs NNMI generated in that 24 hour interval are already in the database, and do not need to be imported at failover time.

The information provided in the above section will help you understand how the network might perform after a failover when using NNMI with application failover using the embedded database.

Configuring NNMI in a High Availability Cluster



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

NNMI provides support for configuring NNMI to run in an HA cluster under one of several separately purchased HA products. Most of the NNM Smart Plug-ins (iSPIs), but not the NNM iSPI NET Diagnostics Server, can also run under HA.

Note: The NNM iSPI NET Diagnostics Server can be installed with NNM iSPI NET and NNMI Ultimate.

Note: When configuring NNMI in a high availability cluster, it is important to follow the standard configuration procedures included in this chapter. Nonstandard configurations are not supported.

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.

Note: Use the NNMi HA commands to ensure the proper configuration of HA for NNMi.

Tip: If you plan to install any NNM iSPiS on the NNMi management server, also see the documentation for those NNM iSPiS.

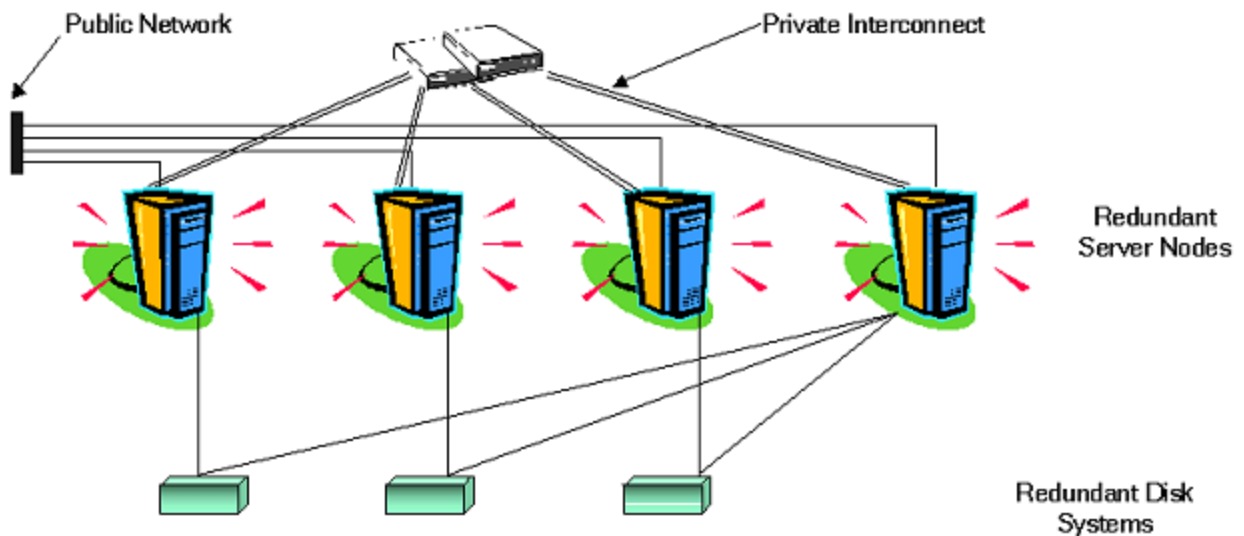
This chapter contains the following topics:

- ["High Availability Concepts" below](#)
- ["Verifying the Prerequisites to Configuring NNMi for High Availability" on page 157](#)
- ["Configure High Availability" on page 159](#)
- ["Shared NNMi Data in High Availability Environments" on page 172](#)
- ["Licensing NNMi in a High Availability Cluster" on page 176](#)
- ["Maintaining the High Availability Configuration" on page 177](#)
- ["Unconfiguring NNMi from an HA Cluster" on page 181](#)
- ["Patching NNMi under HA" on page 184](#)
- ["Troubleshooting the HA Configuration" on page 185](#)
- ["High Availability Configuration Reference" on page 193](#)

High Availability Concepts

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

Architecture of a High Availability Cluster

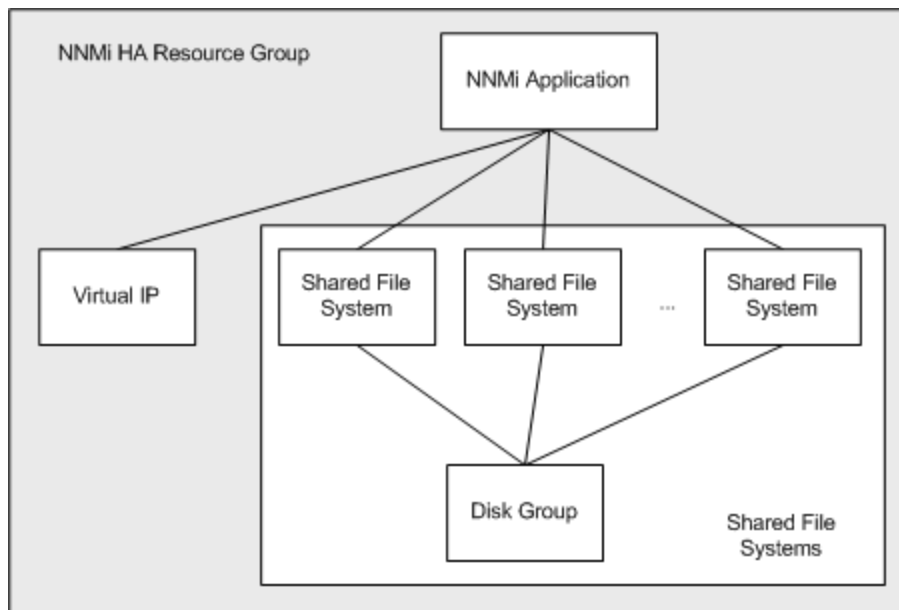


Each node in a cluster connects to one or more public networks and also connects to a private interconnect, representing a communication channel for transmitting data between cluster nodes.

In modern cluster environments such as Veritas Cluster Server, Microsoft Failover Clustering, or Microsoft Cluster Services, applications are represented as compounds of resources, which are simple operations that enable applications to run in a cluster environment. The resources construct an **HA resource group**, which

represents an application running in a cluster environment. The following diagram shows an example High Availability (HA) resource group.

Typical HA Resource Group Layout



This document uses the term *HA resource group* to designate a set of resources in any cluster environment. Each HA product uses a different name for the HA resource group. The following table lists the term for each supported HA product that equates to *HA resource group* for this document. (For the specific supported versions of each HA product, see the NNMi Support Matrix.)

Terminology for HA Resource Group in the Supported HA Products

HA Product	Abbreviation	Equivalent Term for HA Resource Group
Windows Server Failover Clustering	WSFC	Resource Group
Veritas Cluster Server	VCS	Service Group
Red Hat Cluster Suite	RHCS	Service

High Availability Terms

The following table lists and defines some common High Availability (HA) terms.

Common HA Terms

Term	Description
HA resource group	An application running in a cluster environment (under an HA product). An HA resource group can simultaneously be a cluster object that represents an application in a cluster.
Volume group	One or more disk drives that are configured to form a single large storage area.
Logical volume	An arbitrary-size space in a volume group that can be used as a separate file

Common HA Terms, continued

Term	Description
	system or as a device swap space.
Primary cluster node	<p>The first system on which the software product is installed, <i>and</i> the first system on which HA is configured.</p> <p>The shared disk is mounted on the primary cluster node for initial set up.</p> <p>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.</p>
Secondary cluster node	Any system that is added to the HA configuration after the primary cluster node has been fully configured for HA.
Active cluster node	The system that is currently running the HA resource group.
Passive cluster node	Any system that is configured for HA but is not currently running the HA resource group. If the active cluster node fails, the HA resource group fails over to one of the available passive cluster nodes, which then becomes the active cluster node for that HA resource group.

NNMi High Availability Cluster Scenarios

Note: NNMi supports clusters where the application can run on more than two cluster nodes. See the *nms-ha* manpage and the *nnmdatareplicator.ovpl* reference page, or the Linux manpage for more information.

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

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

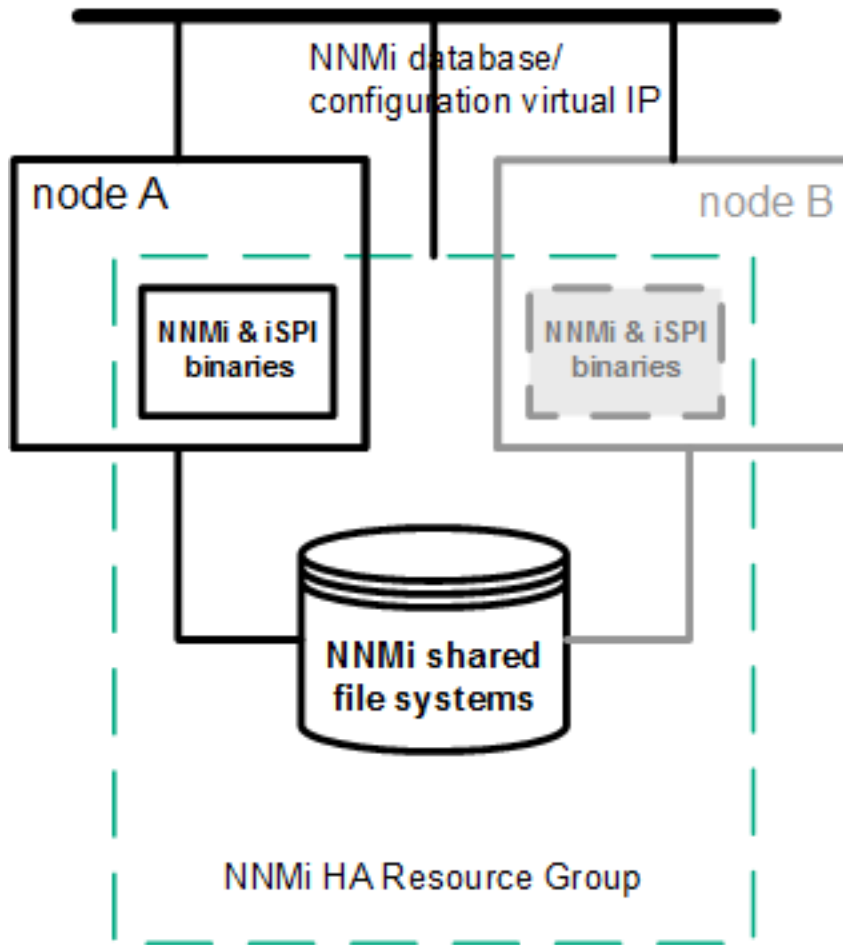
Note: Run the NNMi database backup and restore scripts on the active cluster node only.

NNMi-only scenario

The following diagram shows a graphical representation of the NNMi HA cluster scenario. In this figure the NNMi HA resource group is synonymous with the NNMi HA cluster.

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

If the cluster contains more than two NNMi nodes, additional nodes are configured similarly to node B in the following diagram

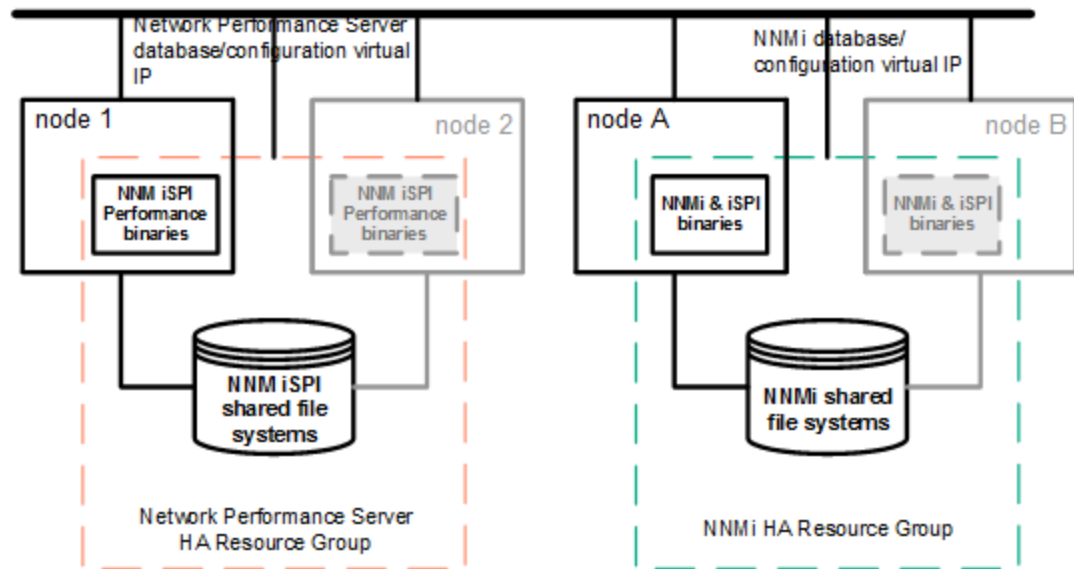
Basic Scenario for NNMi HA Cluster

For information about how to implement this scenario, see [Configure NNMi for High Availability](#) and [Configure NNM iSPI for High Availability](#).

NNMi and NNM Performance iSPIs on a standalone server scenario

If you are running any of the NNM Performance iSPIs on a standalone server, you can configure these NNM iSPIs to run as a separate HA resource group within the NNMi HA cluster, as shown in the following diagram. The NNMi HA resource group is the same as that described for the NNMi-only scenario.

HA for NNMi and NNM Performance iSPIs on a Standalone Server



For information about how to implement this scenario, see [Configure NNMi for High Availability](#) and [Configure NNM iSPI for High Availability](#)

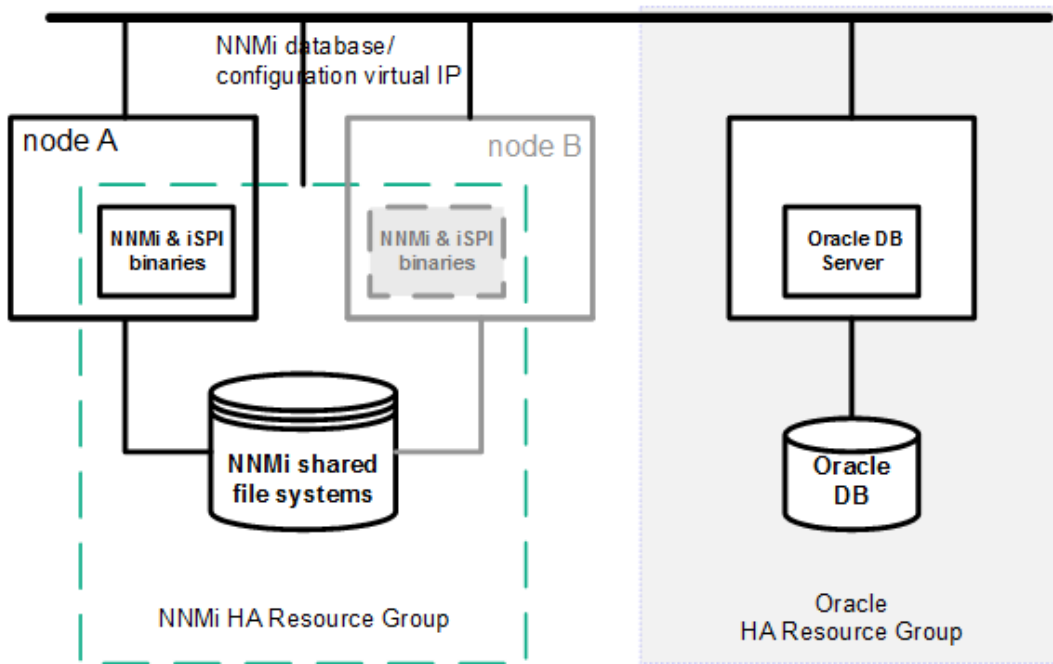
- Other options for the NNM Performance iSPIs on a standalone server are as follows:
- Run the NNM Performance iSPIs on a single system with no HA. Use this approach while evaluating the NNM iSPIs and for environments where it is not critical for performance data to be always available.
- Configure the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

NNMi with an Oracle database scenario

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

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

HA for NNMi with an Oracle Database

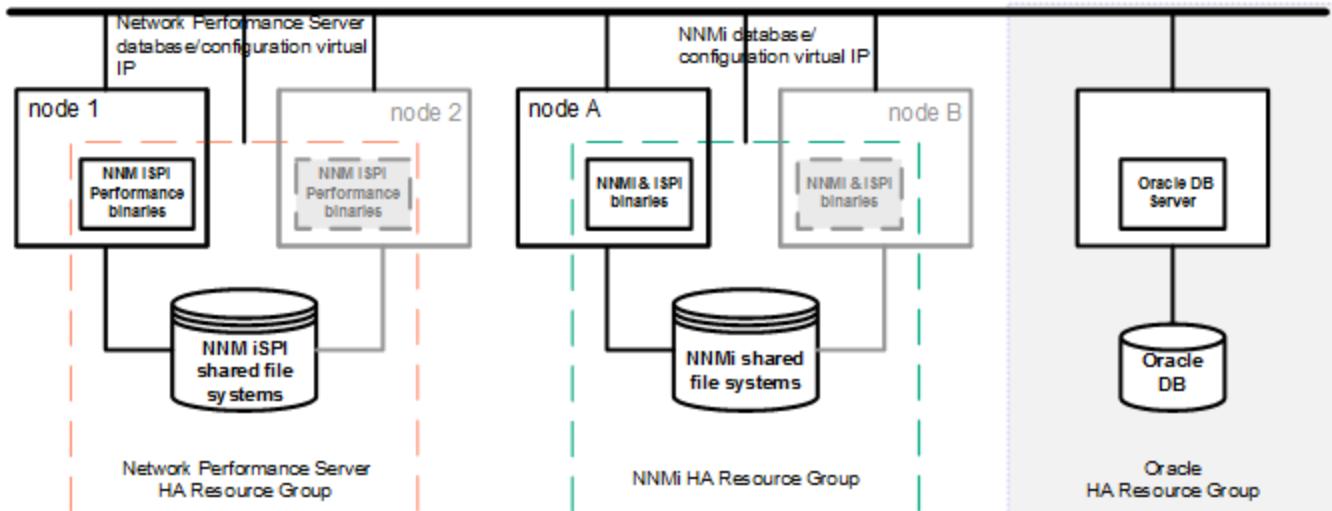


For information about how to implement this scenario, see "Configure NNMi for High Availability in an Oracle Environment" on page 171 and "Configure NNM iSPIs for High Availability" on page 169.

NNMi with an Oracle database and NNM Performance iSPIs on a standalone server scenario

If your NNMi implementation uses Oracle for the main NNMi database and you are running any of the NNM Performance iSPIs on a standalone server, you can configure three HA resource groups within the NNMi HA cluster, as shown in the following diagram.

HA for NNMi with an Oracle Database and NNM Performance iSPIs on a Standalone Server



For information about how to implement this scenario, see "Configure NNMi for High Availability in an Oracle Environment" on page 171 and "Configure NNM iSPIs for High Availability" on page 169.

Manpages

NNMi provides the following manpages to assist you with 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.

Verifying the Prerequisites to Configuring NNMi for High Availability

Successful configuration of NNMi for High Availability (HA) depends on a number of factors:

- Appropriate hardware
- Understanding of the HA product
- A methodical approach to configuration

Before you begin to configure NNMi for HA, complete the following preparation:

1. Verify that NNMi supports your HA product by checking the information in the NNMi Support Matrix.
2. Read the documentation for your HA product to familiarize yourself with the capabilities of that product and to make design decisions.

Tip: HA product documentation changes frequently. Be sure you have the most recent versions available.

3. Verify that each system to be included as a node in an NNMi HA cluster meets the following requirements:
 - Meets all requirements described in the documentation for the HA product.
 - Includes at least two network interface cards (NIC cards).

Note: Review the HA product, operating system, and NIC card documentation to verify that these products can all work together.

- Supports the use of a virtual IP address for the HA resource group. This IP address is the IP address used for the NNMi license.

Note: WSFC requires multiple virtual IP addresses, one for the HA cluster and one for each HA resource group. In this case, the virtual IP address of the NNMi HA resource group is the IP address used for the NNMi license.

- Supports the use of a shared disk or disk array

Note: Review the HA product, operating system, and disk manufacturer documentation to verify that these products, including the related SCSI cards, can all work together.

- Meets all requirements for NNMi as described in the *NNMi Support Matrix*.
4. If you plan to run any NNM iSPIs in the NNMi HA cluster, read the appropriate NNM iSPI documentation for additional HA configuration prerequisites.
 5. Allocate the following virtual IP addresses and host names:
 - One virtual IP address for the HA cluster (WSFC only)
 - One virtual IP address for each HA resource group to be configured
 6. From any system, use the `nslookup` command to validate correct DNS response for all of the IP addresses and hostnames you allocated in [step 5](#).
 7. Verify that operating system of each system is at the correct version and patch level for the HA product and NNMi.
 8. If necessary, install the HA product.
 9. Prepare the shared disk as described in "[Prepare the Shared Disk Manually in High Availability Environments](#)" on page 174.
 10. Use the commands for your HA product to configure (if necessary) and test an HA cluster.

The HA cluster provides such functionality as checking the application heartbeat and initiating failover. The HA cluster configuration must, at a minimum, include the following items:

 - (Linux only) `ssh`, `remsh`, or both
 - (Windows only) Virtual IP address for the HA cluster that is DNS-resolvable
 - Virtual hostname for the HA cluster that is DNS-resolvable
 - A resource group that is unique and specific to NNMi.

Note: NNMi expects that the NNMi HA resource group includes all required resources. If this is not the case, use the HA product functionality to manage dependencies between the NNMi HA resource group and the other HA resource groups. For example, if Oracle is running in a separate HA resource group, configure the HA product to ensure that the Oracle HA resource group is fully started before the HA product starts the NNMi HA resource group.

- *WSFC*: Use the create cluster wizard of Failover Cluster Management for Windows Server.
- *VCS*: Not necessary. Product installation created an HA cluster.
- *RHCS*: Add services (`cman`, `rgmanager`) as described in the RHCS documentation.

For information about testing the resources that you will place into the NNMi HA resource group, see "[HA Resource Testing](#)" on page 186.

Configure High Availability

This section describes the procedures for configuring a new High Availability (HA) configuration for NNMi. It contains the following topics:

- ["Configure NNMi Certificates for High Availability" below](#)
- ["Configure NNMi for High Availability" below](#)
- ["Configure NNM iSPIs for High Availability" on page 169](#)
- ["Configure NNMi for High Availability in an Oracle Environment" on page 171](#)

Note: When configuring HA, note the following general guidelines:

- RHCS configuration requires a complete restart of the HA cluster daemons, including all applications, on each node in the HA cluster. Plan your configuration effort accordingly.
- Do not use the RHCS luci Web interface to change the NNMi resource group. The luci Web interface removes the NNMi resource group global variables from `/etc/cluster/cluster.conf` if changes are made to the NNMi resource group. The NNMi resource group global variables are required for proper NNMi HA functionality.
- By default, in an HA environment, the SNMP source address is set to a physical cluster node address. To set the SNMP source address to the `NNM_INTERFACE` (which is set to the virtual IP address), you must edit the `ov.conf` file and set the value for `IGNORE_NNM_IF_FOR_SNMP` to `OFF`. (By default, this setting is set to `ON`.)
- 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 ["Maintenance Mode" on page 177](#) for more information.

Configure NNMi Certificates for High Availability

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 High Availability (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 ["Working with Certificates in High-Availability Environments" on page 258](#). You can complete this procedure before or after configuring NNMi for HA.

Configure NNMi for High Availability

The two distinct phases of configuring NNMi for High Availability (HA) are as follows:

1. Copy the NNMi data files to the shared disk.
 - Do this task on the primary node, as described in [step 1](#) through [step 9](#) of ["Configuring NNMi on the Primary Cluster Node" on page 165](#).
2. Configure NNMi to run under HA.

- Do this task on the primary node, as described in [step 10](#) through [step 15](#) of "[Configuring NNMi on the Primary Cluster Node](#)" on [page 165](#).
- Also do this task on the secondary node, as described in "[Configuring NNMi on the Secondary Cluster Nodes](#)" on [page 168](#).

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

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

During failover, the NNMi console is unresponsive. After failover completes, NNMi users must log on to continue their NNMi console sessions.

The following diagram provides an illustration of the NNMi HA configuration process.

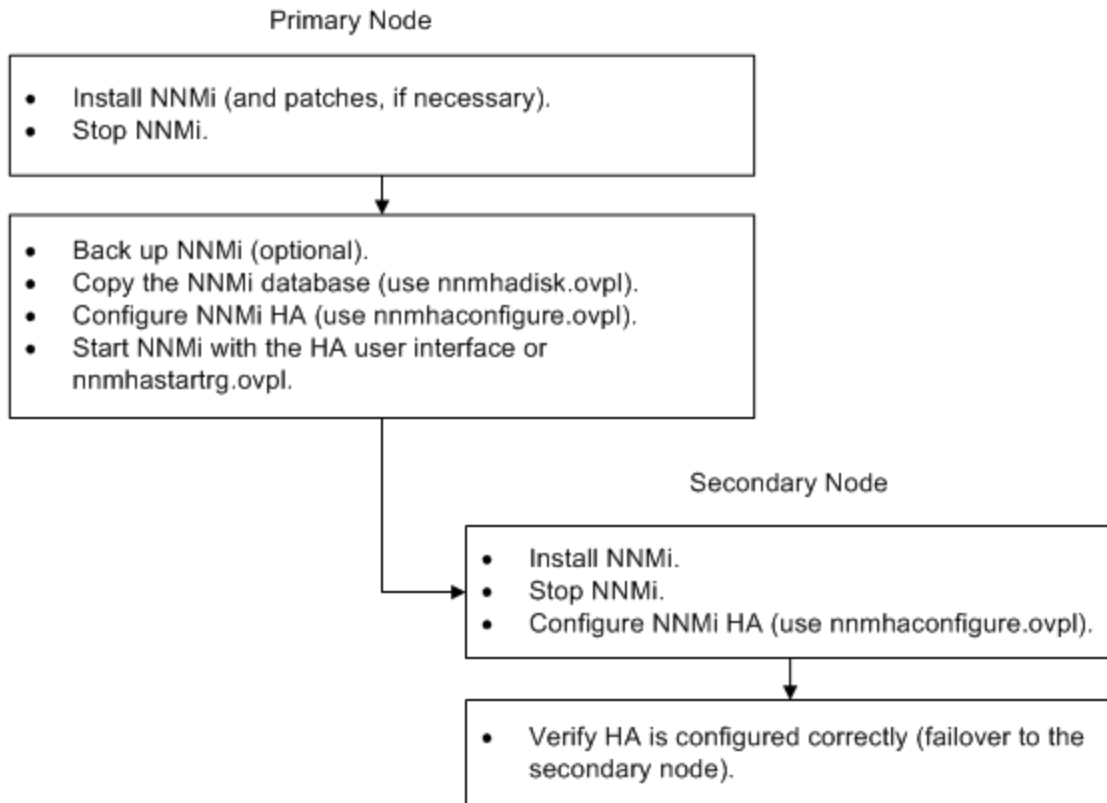
NNMi HA Configuration Workflow

HA Configuration

Configure the cluster on both nodes (primary and secondary), including the shared disk:

- Verify the prerequisites to configure NNMi for HA.
- Set up the HA Cluster according to the operating system vendor documentation.
- Verify that the HA cluster is configured correctly.

NNMi Installation & Configuration



Note: If you encounter errors during HA configuration, do the following:

1. Unconfigure NNMi from the HA environment by running the `nnmhaunconfigure.ovpl` command.
2. Correct the condition indicated by the error message(s).
3. Reconfigure NNMi into the HA environment by running the `nnmhaconfigure.ovpl` command.

(RHCS only) For the `nnmhaconfigure.ovpl` and `nnmhaunconfigure.ovpl` commands to work properly, the `<failoverdomains/>` tag must exist in the `/etc/cluster/cluster.conf` file.

The `<failoverdomains/>` tag is embedded within the resource manager section, for example:

```
...
...
<rm>
```

```

    <failoverdomains/>
</rm>

The nmhaconfigure.ovpl command requires the <failoverdomains/> tag to create the NNMI
resource group, using the following example structure:

...
<rm>
  <failoverdomains>
    <failoverdomain name="<rg-name>-dom" nofailback="0"
ordered="0" restricted="1">
      <failoverdomainnode name="<node1>" priority="1"/>
      <failoverdomainnode name="<node2>" priority="1"/>          </failoverdomain>
    </failoverdomains>
    <service autostart="1" domain="<rg-name>-dom"
exclusive="0" name="nmha" recovery="relocate">
      <ip address="<addr>" monitor_link="1">
        <fs device="<nmhalvol>" force_fsck="1"
force_unmount="1" fsid="" fstype="ext3"
mountpoint="<nnm-hamount>" name="nmha-mount"
options="" self_fence="0">
          <NNMscript GLOBAL_VARIABLES="NNM_INTERFACE=
<virtual hostname>;HA_LOCALE=en_US.UTF-8;
HA_MOUNT_POINT="/<nnm-hamount>"
file="/var/opt/OV/hacluster/<rg-name>/nmharhcs"
name="nmha-APP"/>
        </fs>
      </ip>
    </service>
  </rm>

```

The `nmhaunconfigure.ovpl` command also requires the above structure to remove the node's failoverdomain entry.

For more information, see the `nmhaunconfigure.ovpl` and `nmhaconfigure.ovpl` reference pages, or the Linux manpages.

NNMi High Availability Configuration Information

The High Availability (HA) configuration script collects information about the NNMi HA resource group. Please prepare the information listed in the following table before you configure NNMi HA. This information is needed to execute the HA script (`nnmhaconfigure.ovp1`) interactively, depending on your operating system or HA software.

NNMi HA Primary Node Configuration Information

HA Configuration Item	Description
HA resource group	<p>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. See your HA system provider's reference material for information on valid names.</p> <p>Upon input of an HA resource group name, NNMi generates the following resources for Linux and Windows systems:</p> <pre><resource group name>-IP <resource group name>-Mount <resource group name>-App</pre> <p>In addition, for Windows systems, the following resource is generated upon input of a virtual hostname:</p> <pre><virtual hostname></pre>
Virtual host short name	<p>The short name for the virtual host. This hostname must map to the virtual IP address for the HA resource group. The <code>nslookup</code> command must be able to resolve the virtual host short name and the virtual IP address.</p> <div style="background-color: #e0e0e0; padding: 10px; border: 1px solid #ccc;"> <p>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, HPE recommends that you implement a secondary naming strategy (such as entering the information in the <code>%SystemRoot%\system32\drivers\etc\hosts</code> file on the Windows operating system or <code>/etc/hosts</code> file on UNIX operating systems) in case DNS is not available during NNMi HA configuration.</p> </div>
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	<p>The network interface on which the virtual host IP address is running. For example:</p> <ul style="list-style-type: none"> • <i>Windows:</i> Local Area Connection • <i>Linux:</i> eth0
Shared file system type	The type of shared disk configuration being used for the HA resource group. Possible values are:

NNMi HA Primary Node Configuration Information, continued

HA Configuration Item	Description
	<ul style="list-style-type: none"> • <code>disk</code>—The shared disk is a physically attached disk that uses a standard file system type. The HA configuration script can configure the shared disk. For more information, see the File system type entry in this table. • <code>none</code>—The shared disk uses a configuration other than that described for the <code>disk</code> option, such as NFS. After running the HA configuration script, configure the shared disk as described in "Prepare the Shared Disk Manually in High Availability Environments" on page 174.
File system type	<p>(Linux only) The file system type of the shared disk (if the shared file system type is <code>disk</code>). The HA configuration scripts pass this value to the HA product so that it can determine how to validate the disk.</p> <p>HPE has tested the following shared disk formats:</p> <ul style="list-style-type: none"> • <i>Windows</i>: Basic (see "A Note about Shared Disk Configuration on Windows Server" on page 175); SAN • <i>Linux</i>: <code>ext2</code>, <code>ext3</code>, and <code>vxfs</code> for VCS and RHCS <p>Note: HA products support other file system types. If you use a shared disk format that HPE has not tested, prepare the disk before configuring NNMi to run under HA, and then specify <code>none</code> for the shared file system type while running the NNMi HA configuration script.</p>
Disk information (disk group, volume group, and/or logical volume name, depending on the operating system used)	<p>The name associated with the disk information for the NNMi shared file system.</p> <p>Note: When you create/attach a disk on UNIX platforms, for example, with <code>vxfs</code> or <code>lvm</code>, you create different items, such as: disk group, volume group, logical volume. The names for these items are assigned by the system administrator at the time of creation. NNMi does not enforce any naming conventions. Contact your system administrator for your company's naming information.</p>
Mount point	<p>The directory location for mounting the NNMi shared disk. This mount point must be consistent between systems. (That is, each node must use the same name for the mount point.) For example:</p> <ul style="list-style-type: none"> • <i>Windows</i>: <code>S:\</code> <p>Note: Specify the drive completely. <code>S</code> and <code>S:</code> are unacceptable formats and do not provide access to the shared disk.</p> <ul style="list-style-type: none"> • <i>Linux</i>: <code>/nnmmount</code>

Configuring NNMi on the Primary Cluster Node

Complete the following procedure on the primary cluster node.

Note: If you are using Oracle for the main NNMi database, see "[Configure NNMi for High Availability in an Oracle Environment](#)" on page 171 first.

1. If you have not already done so, complete the procedure for "[Verifying the Prerequisites to Configuring NNMi for High Availability](#)" on page 157.
2. If you have not already done so, install NNMi (including the latest consolidated patch, if any), and then verify that NNMi is working correctly.
3. If you expect to run any NNM iSPIs on this NNMi management server, see "[Configure NNM iSPIs for High Availability](#)" on page 169 before continuing with this procedure.
4. Use the `nnmbackup.ovpl` command, or another database command, to back up all NNMi data. For example:

```
nnmbackup.ovpl -type offline -scope all -target nnmi_backups
```

For more information about this command, see "[NNMi Backup and Restore Tools](#)" on page 222.

5. Define the disk device group (and logical volume), consisting of at least one shared disk for the NNMi HA resource group. For example:
 - *WSFC*: Use Disk Management to configure the disk mount point and format the disk.
 - *VCS*:
Use VSF commands such as `vxdiskadm`, `vxassist`, and `mkfs` to add and initialize the disk, allocate disks by space, and create the logical volume.
 - *RHCS*:
Use LVM commands such as `pvcreate`, `vgcreate`, and `lvcreate` to initialize the disk, create the volume group, and create the logical volume.

Note: NNMi requires RHCS clusters be configured such that the cluster node names specified in the `/etc/cluster/cluster.conf` file must be fully qualified for NNMi to correctly start and stop.

For Linux operating systems, a reference web site is:

<http://www.unixguide.net/unixguide.shtml>

6. Create the directory mount point (for example, `S:\` or `/nnmmount`), and then mount the shared disk:
 - *Windows*: Use the Windows Explorer and Disk Management tool to assign a drive letter.

Caution: Use the Disk Management tool make sure that the shared disk displays **online**. If it displays **reserved**, this indicates 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 is changed to **online**.

- *Linux*:

- Use the `mkdir` and `mount` commands.
- Verify that the shared disk directory mount point has been created with `root` as the user, `sys` as the group, and the permissions set to `755`. For example:

```
ls -l /nmmount
```

Caution: After configuration, the HA product manages disk mounting. Do *not* update the files system table with this mount point.

7. Stop NNMi:

```
ovstop -c
```

Note: If NNMi is already installed on a node that you will include in this HA resource group, also run `ovstop -c` on that node at this time.

8. 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>
```

Note: To prevent database corruption, run this command (with the `-to` option) only one time. For information about alternatives, see ["Re-Enable NNMi for High Availability after All Cluster Nodes are Unconfigured" on page 187](#).

9. (Linux only) Unmount the shared disk and deactivate the disk group:

```
umount <HA_mount_point>
```

```
vgchange -a n <disk_group>
```

10. Verify that NNMi is not running:

```
ovstop -c
```

11. (RHCS only) Perform the following to add the necessary NNMscript resource to the `/usr/share/cluster/cluster.rng` file:

- Save a copy of the `cluster.rng` file.
- Edit the `/usr/share/cluster/cluster.rng` file as follows:
 - Find `<define name="CHILDREN">`.
 - Embed the contents of the file `/opt/OV/misc/nnm/ha/NNMscript.rng` ahead of the statement found in the previous step.
For example go one line above `<define name="CHILDREN">`, and type:

```
:r /opt/OV/misc/nnm/ha/NNMscript.rng
```
 - In the CHILDREN XML block, add the text that is bold in the following:

```
<define name="CHILDREN">
```

```

<zeroOrMore>
  <choice>
    ...
    <ref name="SCRIPT"/>
    <ref name="NNMSCRIPT"/>
    <ref name="NETFS"/>

```

- iv. Save the `cluster.rng` file.
- c. Copy the `/opt/OV/misc/nnm/ha/NNMscript.sh` file to `/usr/share/cluster` and ensure that it has 755 permissions with `root:root` ownership.
- d. Restart the `ccsd` service or reboot.
- e. If you rebooted the system in the previous step, before continuing with the cluster configuration, `stopNNMi`:

```
ovstop -c
```

- f. Verify that NNMi is not running:

```
ovstatus -c
```

12. Configure the NNMi HA resource group:

- *Windows*:

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

- *Linux*:

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

13. (Linux only) By default, NNMi starts in the locale of the user who ran the `nmhaconfigure.ovpl` command. To change the NNMi locale, run the following command:

```
$NnmInstallDir/misc/nnm/ha/nnmhaclusterinfo.ovpl -config NNM -set HA_LOCALE <Locale>
```

14. In [step 12](#), determine the value you specified for the shared file system type:

- For type `disk`, the `nmhaconfigure.ovpl` command configured the shared disk. Continue with [step 15](#).
- For type `none`, prepare the shared disk as described in "[Prepare the Shared Disk Manually in High Availability Environments](#)" on [page 174](#), and then continue with [step 15](#).

15. 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>
```

If NNMi does not start correctly, see "[Troubleshooting the HA Configuration](#)" on [page 185](#).

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

Configuring NNMi on the Secondary Cluster Nodes

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

1. If you have not already done so, complete the procedure for "[Configuring NNMi on the Primary Cluster Node](#)" on page 165.
2. If you have not already done so, complete the procedure for "[Verifying the Prerequisites to Configuring NNMi for High Availability](#)" on page 157.
3. If you have not already done so, install NNMi (including the latest consolidated patch, if any), and then verify that NNMi is working correctly.
4. Install the NNM iSPiS that you installed in [step 3](#) of "[Configuring NNMi on the Primary Cluster Node](#)" on page 165.
5. Stop NNMi:
ovstop -c
6. Create a mount point for the shared disk (for example, `S:\` or `/nnmmount`).

Note: This mount point must use the same name as the mount point you created in [step 6](#) of the procedure "[Configuring NNMi on the Primary Cluster Node](#)" on page 165.

7. (RHCS only) Perform the following to add the necessary NNMscript resource to the `/usr/share/cluster/cluster.rng` file:
 - a. Save a copy of the `cluster.rng` file.
 - b. Edit the `/usr/share/cluster/cluster.rng` file as follows:
 - i. Find `<define name="CHILDREN">`
 - ii. Embed the contents of the file `/opt/OV/misc/nm/ha/NNMscript.rng` ahead of the statement found in the previous step.
For example go one line above `<define name="CHILDREN">`, and type:

```
:r /opt/OV/misc/nm/ha/NNMscript.rng
```
 - iii. In the CHILDREN XML block, add the text that is bold in the following:

```
<define name="CHILDREN">
  <zeroOrMore>
    <choice>
      ...
      <ref name="SCRIPT"/>
      <ref name="NNMSCRIPT"/>
      <ref name="NETFS"/>
    </choice>
  </zeroOrMore>
</define>
```
 - iv. Save the `cluster.rng` file.
8. (RHCS only) Copy the NNMi custom script into place, and then restart the HA cluster daemons.

- a. Copy the `/opt/OV/misc/nnm/ha/NNMscript.sh` file to the following location:
`/usr/share/cluster/NNMscript.sh`
 - b. Stop and then restart the `/sbin/ccsd` process.
9. Configure the NNMi HA resource group:
 - *Windows:* `%NnmInstallDir%\misc\nnm\ha\nnmhaconfigure.ovpl NNM`
 - *Linux:* `$NnmInstallDir/misc/nnm/ha/nnmhaconfigure.ovpl NNM`

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

10. Verify that the configuration was successful:
 - *Windows:*
`%NnmInstallDir%\misc\nnm\ha\nnmhaclusterinfo.ovpl
 -group <resource_group> -nodes`
 - *Linux:*
`$NnmInstallDir/misc/nnm/ha/nnmhaclusterinfo.ovpl
 -group <resource_group> -nodes`

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

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

Configure NNM iSPIs for High Availability

If you expect to run any NNM iSPIs on the NNMi management server, read this section before configuring NNMi to run under HA.

NNM iSPI Performance for Metrics, NNM iSPI Performance for QA, and NNM iSPI Performance for Traffic

The NNM iSPI Performance for Metrics can be installed on the NNMi management server or on a standalone server.

The NNM iSPI Performance for Traffic has two different components (Traffic Master and Traffic Leaf), which can be installed on the NNMi management server or standalone servers, or a combination of both (one component on the NNMi management server and the other on a remote server).

Note:

- If the NNM iSPI (or component) will be located on the NNMi management server, install the product before configuring NNMi to run under HA.
- If the NNM iSPI (or component) will be located on a standalone server, configure NNMi to run under HA before installing the product. During the NNM iSPI installation process, supply the NNMi HA resource group virtual hostname as the NNMi management server name.

For more information on installing an NNM iSPI, see the appropriate NNM iSPI installation guide.

NNM iSPI Performance for QA, NNM iSPI for MPLS, NNM iSPI for IP Multicast, and NNM iSPI for IP Telephony

The NNM iSPI Performance for QA, NNM iSPI for MPLS, NNM iSPI for IP Multicast, and NNM iSPI for IP Telephony can be installed on the NNMi management server only.

For information about configuring the NNM iSPIs to run under HA, see the documentation for the appropriate NNM iSPI.

NNM iSPI Network Engineering Toolset Software and NNMi Running under HA

The NNM iSPI Network Engineering Toolset Software SNMP trap analytics and Microsoft Visio export functionality are automatically installed with the NNMi Premium or NNMi Ultimate products. No extra work is needed to run these tools under HA.

The NNM iSPI NET Diagnostics Server cannot be included in the NNMi HA resource group. Do not install this component on the NNMi management server. To run the NNM iSPI NET Diagnostics Server on a system that is outside the NNMi HA resource group, follow these steps:

Note: The NNM iSPI NET Diagnostics Server requires an NNM iSPI NET or NNMi Ultimate license. See the *HPE NNM iSPI Network Engineering Toolset Software Interactive Installation and Upgrade Guide* for information about how to install and configure this server.

1. Completely configure the NNMi HA resource group.
2. Install the NNM iSPI NET Diagnostics Server on a system that is outside the NNMi HA resource group. During the NNM iSPI NET Diagnostics Server installation process, supply the NNMi HA resource group virtual hostname as the NNM Server Hostname.

For more information, see the *NNM iSPI Network Engineering Toolset Software Planning and Installation Guide*.

If the NNM iSPI NET Diagnostics Server is already installed on an NNMi management server that will run under HA, uninstall the NNM iSPI NET Diagnostics Server before configuring NNMi to run under HA.

Caution: Uninstalling the NNM iSPI NET Diagnostics Server removes all existing reports.

Note: It might be possible to save existing reports, as described here, but the following procedure is untested:

1. Use MySQL Workbench to perform a backup of the existing `nnminet` database.
MySQL Workbench is available in the downloads area at dev.mysql.com.
2. Uninstall the NNM iSPI NET Diagnostics Server.
3. Configure NNMi to run under HA.
4. Install the NNM iSPI NET Diagnostics Server on a separate system.
5. Before running any flows, use MySQL Workbench to recover the `nnminet` database onto the new installation.

Configure NNMi for High Availability in an Oracle Environment

This sections presents a high-level overview of the process for configuring NNMi with an Oracle database to run under High Availability (HA).

Note: The number of possible Oracle configurations is large, and the configuration process can vary according to the Oracle release. For the most accurate information about configuring Oracle to run under HA and creating an NNMi dependency on the Oracle HA resource group, see the HA product documentation. You can also go to the Oracle web site (www.oracle.com) for information about the appropriate Oracle configuration for your HA product.

NNMi Dependency on Oracle in High Availability Environments

When Oracle and NNMi both run under High Availability (HA), the NNMi HA resource group must include a shared disk for the NNMi data that is not stored in the Oracle database.

Additionally, consider the following information:

- If the HA product supports dependencies, the recommended approach is to configure each product to run in a separate HA resource group. The Oracle HA resource group must be fully started before the NNMi HA resource group starts. If both HA resource groups are in the same HA cluster, you can modify the cluster configuration to set resource group ordering. If the HA resource groups are in different HA clusters, make sure that the NNMi HA resource group dependency on the Oracle HA resource group is met.
- If the HA product does not support dependencies, include the Oracle systems and the NNMi systems in the NNMi HA resource group.

Configuring NNMi for High Availability in an Oracle Environment

1. If you plan to run Oracle under High Availability (HA), complete that configuration first.
2. Create an empty Oracle database instance for NNMi.
3. On the primary NNMi node, install NNMi (including the latest consolidated patch, if any). During installation, do the following:
 - a. Select the **Oracle** database type, and then select **Primary Server Installation**.
 - b. Specify the virtual IP address or hostname for the Oracle HA resource group (if applicable).
4. On the primary NNMi node, configure NNMi to run under HA as described in "[Configuring NNMi on the Primary Cluster Node](#)" on page 165.
5. Set up the NNMi dependency on the Oracle HA resource group.
For specific instructions, see the HA product documentation.
6. On the secondary NNMi node, install NNMi (including the latest consolidated patch, if any). During installation, do the following:
 - Select the **Oracle** database type, and then select **Secondary Server Installation**.
 - Specify the virtual IP address or hostname for the Oracle HA resource group (if applicable).
7. On the secondary NNMi node, configure NNMi to run under HA described in "[Configuring NNMi on the Secondary Cluster Nodes](#)" on page 168.
8. For each additional secondary NNMi node, repeat [step 6](#) and [step 7](#).

Shared NNMi Data in High Availability Environments

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

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

Data on the NNMi Shared Disk in High Availability Environments

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

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

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

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

- *Windows:*
 - %NnmDataDir%\shared\nnm\databases\Postgres
The embedded database; not present when using an Oracle database.
 - %NnmDataDir%\log\nnm
The NNMi log directory.
 - %NnmDataDir%\nmsas\NNM\log
The NNMi audit log directory.
 - %NnmDataDir%\nmsas\NNM\conf
The NNMi directory for configuring the audit log file.
 - %NnmDataDir%\nmsas\NNM\data
The transactional store used by ovjboss.
- *Linux:*
 - \$NnmDataDir/shared/nnm/databases/Postgres
The embedded database; not present when using an Oracle database.
 - \$NnmDataDir/log/nnm
The NNMi log directory.

- `%NnmDataDir/nmsas/NNM/log`
The NNMi audit log directory.
- `%NnmDataDir/nmsas/NNM/conf`
The NNMi directory for configuring the audit log file.
- `$NnmDataDir/nmsas/NNM/data`
The transactional store used by ovjboss.

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

Replication of Configuration Files in High Availability Environments

The NNMi High Availability (HA) implementation uses file replication to maintain copies of the NNMi configuration files on all NNMi nodes in the HA cluster.

By default, NNMi manages file replication, copying NNMi configuration files from the active node to a passive node during the failover process. The `nmdatareplicator.conf` file specifies the NNMi folders and files included in data replication.

Disabling Data Replication

You can disable data replication as follows:

1. Edit the following file:
 - *Windows:* `%NnmDataDir%\shared\nnm\conf\ov.conf`
 - *Linux:* `$NnmDataDir/shared/nnm/conf/ov.conf`

2. Include the following line:

```
DISABLE_REPLICATION=DoNotReplicate
```

3. Save your changes.

Note: When you change files (for example, configuration files) on the Active node, these files are automatically replicated to the Standby node on failover.

4. Restart the NNMi management server:

Note: 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 "[Maintenance Mode](#)" on page 177 for more information.

- a. Run the `ovstop` command on the NNMi management server.
- b. Run the `ovstart` command on the NNMi management server.

Prepare the Shared Disk Manually in High Availability Environments

If the shared disk is of a format that is supported by HPE, the High Availability (HA) configuration script prepares the shared disk, and you can ignore this section. See "[NNMi High Availability Configuration Information](#)" on page 163 for more information about supported disk formats.

If the shared disk uses a non-tested configuration, such as disk formats supported by the HA product, you must prepare the disk manually. Enter the value none for the file system type during HA configuration, and then configure the shared disk and the NNMi HA resource group's use of the shared disk.

Tip: You can configure the disk before or after configuring the NNMi HA resource group.

To prepare the shared disk manually, follow these steps:

1. Configure the shared disk as described in "[Configuring a SAN or a Physically Connected Disk](#)" below.
2. Configure the NNMi HA resource group to recognize the disk by completing both of the following procedures:
 - "[Setting the High Availability Variables in the ov.conf File](#)" on the next page
 - "[Moving the Shared Disk into the NNMiHA Resource Group](#)" on the next page

Configuring a SAN or a Physically Connected Disk

Connecting and formatting a disk that disk into a vxfs or ext3 file system. To configure a SAN or a physically-connected disk, follow these steps:

1. Verify that the shared disk is *not* configured to be mounted at system boot time.
The resource group is responsible for mounting the shared disk.
2. Connect the device:
 - For a SAN disk, add the SAN device to the network.
The logical volume on the SAN disk should be in exclusive mode, if that mode is available.
 - For a physically-connected disk, attach the disk using a Y cable.
3. Add operating system entries to all cluster nodes (disk group, logical volume, volume group, and disk):
 - For a SAN disk, the entries reference the SAN.
 - For a physically-connected disk, the entries reference the disk hardware.
4. Format the disk using a supported disk format. See "[NNMi High Availability Configuration Information](#)" on page 163 for more information.
5. Ensure that the SAN mounts.

Tip: For Linux systems, a reference web site is: <http://www.unixguide.net/unixguide.shtml>

6. Unmount and deport the disk.
7. To test the configuration, add the disk to a resource group and initiate failover.

Setting the High Availability Variables in the ov.conf File

The NNMi High Availability (HA) resource group uses the following variables to access the shared disk:

- HA_POSTGRES_DIR=<HA_mount_point>/NNM/dataDir/shared/nnm/databases/Postgres
- HA_EVENTDB_DIR=<HA_mount_point>/NNM/dataDir/shared/nnm/eventdb
- HA_NNM_LOG_DIR=<HA_mount_point>/NNM/dataDir/log
- HA_JBOSS_DATA_DIR=<HA_mount_point>/NNM/dataDir/nmsas/NNM/data
- HA_MOUNT_POINT=<HA_mount_point>
- HA_CUSTOMPOLLER_DIR=<HA_mount_point>/NNM/dataDir/shared/nnm/databases/custompoller

Tip: If you plan to run any NNM iSPIs in the NNMi HA resource group, also set the ov.conf variables for each of those NNM iSPIs. For more information, see the documentation for the appropriate NNM iSPI.

To set the product variables for accessing the shared disk in the ov.conf file, run the following command for each of the preceding variables:

- *Windows:*

```
%NnmInstallDir%\misc\nnm\ha\nnmhaclusterinfo.ovpl -config NNM -set <variable> <value>
```

- *Linux:*

```
$NnmInstallDir/misc/nnm/ha/nnmhaclusterinfo.ovpl -config NNM -set <variable> <value>
```

Note: 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 ["Maintenance Mode" on page 177](#) for more information.

Moving the Shared Disk into the NNMiHA Resource Group

Modify the disk configuration file according to the product documentation to move the shared disk into the NNMi HA resource group. For example:

Tip: You can also use this process to add other resources, such as a NIC card or a backup disk to the NNMi HA resource group.

- *WSFC:* Use Failover Management to add resources to the resource group.
- *VCS:* Add disk entries and links to the HA configuration file by using the /opt/VRTSvcs/bin/hares command. For example:
- *RHCS:*

```
/etc/cluster/cluster.conf
```

A Note about Shared Disk Configuration on Windows Server

Note: According to Microsoft Knowledge Base article 237853, dynamic disks are not supported for

clustering with Windows Server.

To ensure the correct disk configuration, review the information located on the following web sites:

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

Licensing NNMi in a High Availability Cluster

NNMi requires two licenses to run NNMi in a High Availability (HA) cluster:

- One production license locked to the IP address of one of the physical cluster nodes
- One non-production license locked to the virtual IP address of the NNMi HA resource group

The NNMi license keys are managed on the shared disk. Therefore, each NNMi HA resource group requires only the non-production license keys for each separately licensed product.

When licensing NNMi in an HA cluster, you must update the licenses.txt file on the shared disk with the new information from the license file on the active node. Complete the following procedure to correctly license NNMi in an HA cluster.

- If you have purchased NNMi Premium or NNMi Ultimate, instead of using a non-production license as directed, you need to use the license key or license keys you requested from the HPE Password Delivery Center for use with High Availability. Obtain a license key for the virtual IP address of the NNMi HA resource group. This license key is initially used on the primary server and then used on the secondary server when needed.

Caution: Do not use production and non-production licenses on the same server.

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

1. Obtain and install a permanent license key for each of your ordered products as described in "[Apply Licenses](#)" on page 241. When prompted for the IP address of the NNMi management server, provide the virtual IP address of the NNMi HA resource group.
2. Update the licenses.txt file on the shared disk with the new information from the LicFile.txt file on the active node. Do one of the following:
 - If the licenses.txt file exists in the NNM directory on the shared disk, append the new license keys in LicFile.txt on the active node to licenses.txt on the shared disk.
 - If the licenses.txt file does not exist on the shared disk, copy LicFile.txt from the active node to licenses.txt in the NNM directory on the shared disk.

On the active node, the LicFile.txt file is in the following location:

- *Windows:* %NnmDataDir%\shared\nnm\conf\licensing\LicFile.txt
- *Linux:* \$NnmDataDir/shared/nnm/conf/licensing/LicFile.txt

On the shared disk, example locations of the licenses.txt file are as follows:

- *Windows:* S:\NNM\licenses.txt
- *Linux:* /nnmount/NNM/licenses.txt

Maintaining the High Availability Configuration

This section describes how to perform the following High Availability configuration maintenance tasks:

["Maintenance Mode" below](#)

["Maintaining NNMi in an HA Cluster" on the next page](#)

["Maintaining Add-on NNM iSPIs in an NNMi HA Cluster" on page 181](#)

Maintenance Mode

When you need to apply NNMi patches or update to a newer version of NNMi, put the NNMi HA resource group into maintenance mode to prevent failover during the process. When the NNMi HA resource group is in maintenance mode, you (or an installation script) can run the `ovstop` and `ovstart` commands as needed on the primary (active) cluster node.

Caution: Never run the `ovstart` or `ovstop` commands on a secondary (backup) cluster node.

Putting an HA Resource Group into Maintenance Mode

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

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

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

Note: The maintenance file contents are as follows:

- To disable monitoring of the HA resource group, create the maintenance file. The file can be empty or can contain the keyword `NORESTART`.
- To prevent NNMi from starting during a configuration procedure, the first line of the maintenance file must contain only the single word:
`NORESTART`

Removing an HA Resource Group from Maintenance Mode

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

To remove an HA resource group from maintenance mode, follow these steps:

1. Verify that NNMi is running correctly:
ovstatus -c
All NNMi services should show the state `RUNNING`.

2. Delete the maintenance file from the node that was the active cluster node before maintenance was initiated. This file is described in ["Putting an HA Resource Group into Maintenance Mode" on the previous page.s](#)

Maintaining NNMi in an HA Cluster

This section describes how to perform the following tasks that might be required to maintain NNMi in a High Availability (HA) Cluster.

["Starting and Stopping NNMi" below](#)

["Changing NNMi Hostnames and IP Addresses in a Cluster Environment" below](#)

["Stopping NNMi Without Causing Failover" on page 180](#)

["Restarting NNMi after Maintenance" on page 181](#)

Starting and Stopping NNMi

Note: While NNMi is running under High Availability (HA), *do not* use the `ovstart` and `ovstop` commands unless instructed to do so for HA maintenance purposes.

For normal operation, use the NNMi-provided HA commands or the appropriate HA product commands for starting and stopping HA resource groups.

Changing NNMi Hostnames and IP Addresses in a Cluster Environment

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

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

- `/etc/hosts`
- Domain Name Service (DNS)
- Network Information Service (NIS)

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

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

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

When changing the virtual hostname or IP address of the NNMi HA resource group, you must update the `licenses.txt` file on the shared disk with the new information from the license file on the active node. Complete the following procedure to correctly update the HA configuration.

To change the virtual hostname or IP address of the NNMi HA resource group, perform these steps on the active NNMi cluster node:

Note: If you have purchased NNMi Premium or NNMi Ultimate, you need to use the license keys you requested from the HPE Password Delivery Center for use with application failover or high availability. Be sure to request the following:

- High Availability: Obtain a license key for the virtual IP address of the NNMi HA resource group. This license key is initially used on the primary server and then used on the secondary server when needed.
- Application Failover: Obtain two license keys; one for the physical IP address of the primary server and one for the physical IP address of the standby server.

1. Convert the license keys for the prior virtual IP address of the NNMi HA resource group to the new virtual IP address of the NNMi HA resource group.

Caution: Do *not* install the new license keys at this time.

2. Put the NNMi HA resource group into maintenance mode as described in "[Putting an HA Resource Group into Maintenance Mode](#)" on page 177.

3. Stop the NNMi HA resource group:

- *Windows:*

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

- *Linux:*

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

4. Change the IP address or node name of the NNMi HA resource group:

- a. In the `ov.conf` file, edit the `NNM_INTERFACE` entry to be the new hostname or IP address.
- b. In the `ovspmd.auth` file, edit any lines containing the old hostname to contain the new hostname.

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

- *Windows:* %NnmDataDir%\shared\nnm\conf

- *Linux:* \$NnmDataDir/shared/nnm/conf

5. If you changed the node name of the NNMi HA resource group, set NNMi to use the new fully-qualified domain name of the NNMi HA resource group with the `nnmsetofficialfqdn.ovpl` command. For example:

```
nnmsetofficialfqdn.ovpl newnnmi.servers.example.com
```

For more information, see the `nnmsetofficialfqdn.ovpl` reference page, or the Linux manpage.

6. Change the cluster configuration to use the new IP address:

- *WSFC:*

In Failover Cluster Management, open `<resource_group>`.

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

- *VCS:*

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

- *RHCS*:

On the active HA cluster node, edit the `/etc/cluster/cluster.conf` file to replace `ip address=<old_IP_address>` with `ip address=<new_IP_address>`. Then run `ccs_tool update /etc/cluster/cluster.conf` to update all other systems.

7. Install the license keys for the new virtual IP address of the NNMi HA resource group as described in ["Apply Licenses" on page 241](#).
8. Update the `licenses.txt` file on the shared disk with the new information from the `LicFile.txt` file on the active node. Do one of the following:
 - If the `licenses.txt` file exists in the NNM directory on the shared disk, append the new license keys in `LicFile.txt` on the active node to `licenses.txt` on the shared disk.
 - If the `licenses.txt` file does not exist on the shared disk, copy `LicFile.txt` from the active node to `licenses.txt` in the NNM directory on the shared disk.

On the active node, the `LicFile.txt` file is in the following location:

- *Windows*: `%NnmDataDir%\shared\nnm\conf\licensing\LicFile.txt`
- *Linux*: `$NnmDataDir/shared/nnm/conf/licensing/LicFile.txt`

On the shared disk, example locations of the `licenses.txt` file are as follows:

- *Windows*: `S:\NNM\licenses.txt`
- *Linux*: `/nnmount/NNM/licenses.txt`

9. 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>
```

10. Verify that NNMi started correctly:

```
ovstatus -c
```

All NNMi services should show the state RUNNING.

11. Take the NNMi HA resource group out of maintenance mode as described in ["Removing an HA Resource Group from Maintenance Mode" on page 177](#).

Stopping NNMi Without Causing Failover

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

Follow these steps on the active cluster node:

1. Put the NNMi HA resource group into maintenance mode as described in ["Putting an HA Resource Group into Maintenance Mode" on page 177](#).
2. Stop NNMi:


```
ovstop -c
```

Restarting NNMI after Maintenance

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

1. Start NNMI:
`ovstart -c`
2. Verify that NNMI started correctly:
`ovstatus -c`
All NNMI services should show the state RUNNING.
3. Take the NNMI HA resource group out of maintenance mode as described in ["Removing an HA Resource Group from Maintenance Mode" on page 177](#).

Maintaining Add-on NNM iSPIs in an NNMI HA Cluster

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

Unconfiguring NNMI from an HA Cluster

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

Note: Before uninstalling NNMI, remove any NNMI patches in reverse order, beginning with the most recent patch. The patch removal process varies according to the operating system running on the NNMI management server. See the patch documentation for installation and removal instructions.

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

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

1. Determine which node in the HA cluster is active. On any node, run the following command:
 - *Windows:*
`%NnmInstallDir%\misc\nnm\ha\nnmhaclusterinfo.ovpl -group <resource_group> -activeNode`
 - *Linux:*
`$NnmInstallDir/misc/nnm/ha/nnmhaclusterinfo.ovpl -group <resource_group> -activeNode`
2. On each passive node, unconfigure any add-on NNM iSPIs from the HA cluster.
For information, see the documentation for each NNM iSPI.
3. On any node in the HA cluster, verify that the add-on NNM iSPIs on all passive nodes have been unconfigured from the HA cluster:

- *Windows:*

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

- *Linux:*

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

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

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

At this time, only the active node hostname should appear in the output. If a passive node hostname appears in the output, repeat [step 2](#) until this command output includes only the active node hostname.

4. On the active node, unconfigure any add-on NNM iSPIs from the HA cluster.

For information, see the documentation for each NNM iSPI. On any node in the HA cluster, verify that the add-on NNM iSPIs on all nodes have been unconfigured from the HA cluster:

- *Windows:*

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

- *Linux:*

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

If any hostname appears in the output, repeat this step until this command output indicates that no iSPIs are configured.

5. On each passive 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 access to the shared disk but does not unconfigure the disk group or the volume group.

6. On each passive node, move the NNMi HA resource group-specific files to a separate location for safe-keeping:

```
%NnmDataDir%\hacluster\<resource_group>\ folder.
```

Tip: If you do not plan to reconfigure the NNMi HA resource group, you do not need to save a copy of these files.

7. On the active node, stop the NNMi HA resource group:

- *Windows:*

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

- *Linux:*

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

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

8. On the active 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 access to the shared disk but does not unconfigure the disk group or the volume group.

9. On the active node, move the NNMi HA resource group-specific files to a separate location for safe-keeping:

```
%NnmDataDir%\hacluster\<resource_group>\ folder
```

Tip: If you do not plan to reconfigure the NNMi HA resource group, you do not need to save a copy of these files.

10. Unmount the shared disk.

- If you want to reconfigure the NNMi HA cluster at some point, you can keep the disk in its current state.
- If you want to use the shared disk for another purpose, copy all data that you want to keep (as described in ["Running NNMi Outside HA with the Existing Database" below](#)), and then use the HA product commands to unconfigure the disk group and volume group.

Running NNMi Outside HA with the Existing Database

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

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

```
ovstop
```

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

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

```
ovstop
```

Caution: To prevent data corruption, make sure that no instance of NNMi is running and accessing

the shared disk.

3. (Linux only) Activate the disk group, for example:

```
vgchange -a e <disk_group>
```

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

- *Windows*: Use Server Manager—>Disk Management.
- *Linux*: `mount /dev/vgnnm/lvnnm /nnmmount`

5. Copy the NNMI files from the shared disk to the local disk:

- *Windows*:

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

- *Linux*:

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

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

- *Windows*: Use Windows Explorer.
- *Linux*: `umount /nnmmount`

7. (Linux only) Deactivate the disk group, for example:

```
vgchange -a n <disk_group>
```

8. Obtain and install the permanent production license keys for the physical IP address of this NNMI management server as described in ["Apply Licenses" on page 241](#).

9. Start NNMI:

```
ovstart -c
```

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

Patching NNMI under HA

To apply a patch for NNMI, work in High Availability (HA) maintenance mode. Follow these steps:

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

- *Windows*:

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

- *Linux*:

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


2. On each passive node, put the NNMi HA resource group into maintenance mode as described in ["Putting an HA Resource Group into Maintenance Mode" on page 177](#).

Include the NORESTART keyword.

3. On each passive node, apply the appropriate patches.

Caution: Never run the `ovstart` or `ovstop` commands on a secondary (backup) cluster node.

4. On all passive nodes, take the NNMi HA resource group out of maintenance mode as described in ["Removing an HA Resource Group from Maintenance Mode" on page 177](#).
5. Fail over to a passive node.
6. Go to the node that was previously active (in [step 1](#)), and then follow these steps:
 - a. Put the NNMi HA resource group of the node into maintenance mode as described in ["Putting an HA Resource Group into Maintenance Mode" on page 177](#).
Include the NORESTART keyword.
 - b. On the node, apply the appropriate patches.

Caution: Never run the `ovstart` or `ovstop` commands on a secondary (backup) cluster node.

- c. On the node, take the NNMi HA resource group out of maintenance mode as described in ["Removing an HA Resource Group from Maintenance Mode" on page 177](#).

Troubleshooting the HA Configuration

This section includes the following topics:

Common High Availability Configuration Mistakes

Some common High Availability (HA) configuration mistakes are listed here:

- Incorrect disk configuration
 - VCS: If a resource cannot be probed, the configuration is somehow wrong. 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 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: Bad network configuration
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 automatically restart at boot time.
Review the HA product documentation for information about how to configure automatic restart on boot up.
- Adding NFS or other access to the OS directly (resource group configuration should be managing this).
- Being in the shared disk mount point during a failover or offlining of the HA resource group.

HA kills any processes that prevent the shared disk from being unmounted.

- Reusing the HA cluster virtual IP address as the HA resource virtual IP address (works on one system and not the other)
- Timeouts are too short. If the products are misbehaving, HA product might time out the HA resource and cause a failover.

WSFC: In Failover Cluster Management, check the value of the **Time to wait for resource to start** setting. NNMi sets this value to 15 minutes. You can increase the value.

- Not using maintenance mode

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 afterwards, use the maintenance mode to keep the resource group online to see what is failing.

- Not reviewing cluster logs (cluster logs can show many common mistakes).

Configuration Issues with RHCS 6

It is possible for the `/etc/cluster/cluster.conf` file versions to differ between the two systems in an HA environment if the `ricci` service is down or has been intentionally disabled. Therefore, monitor the `cluster.conf` file regularly to ensure that the file versions are synchronized.

If the `cluster.conf` file versions are not synchronized, you may experience problems when you attempt to do any of the following:

- apply changes to `cluster.conf`
- unconfigure a resource group
- start the cluster
- use the `clustat` command

HA Resource Testing

This section describes the general approach for testing the resources that you will place into the NNMi HA resource group. This testing identifies hardware configuration problems. It is recommended to perform this testing *before* configuring NNMi to run under High Availability (HA). Note the configuration values that generate positive results, and use these value when performing the complete 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, follow these steps:

1. If necessary, start the HA cluster.
2. (Windows only) Verify that the following virtual IP addresses have been defined for the HA cluster:
 - A virtual IP address for the HA cluster
 - A virtual IP address for each HA resource group

Each of these IP addresses should not be used 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:
 - a. Add the virtual IP address and corresponding virtual hostname 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.
 - b. Fail over from the active cluster node to the passive cluster node to verify that the HA cluster correctly fails over.
 - c. Fail over from the new active cluster node to the new passive cluster node to verify failback.
 - d. If the resource group does not fail over correctly, log on to the active node, and then verify that the IP address is properly configured and accessible. Also verify that no firewall blocks the IP address.
5. Configure the shared disk as described in ["Configuring a SAN or a Physically Connected Disk" on page 174](#).
6. Test the connection to the shared disk:
 - a. Add the shared disk as a resource to the HA resource group as described in ["Moving the Shared Disk into the NNMiHA Resource Group" on page 175](#).
 - b. Fail over from the active cluster node to the passive cluster node to verify that the HA cluster correctly fails over.
 - c. Fail over from the new active cluster node to the new passive cluster node to verify failback.
 - d. If the resource group does not fail over correctly, log on to the active node, and then verify that the disk is mounted and available.
7. 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.
8. Remove the resource group from each node:
 - a. Remove the IP address entry.
 - b. Offline the resource group, and then remove resource group from the node.

At this point, you can use the NNMi-provided tools to configure NNMi to run under HA.

Re-Enable NNMi for High Availability after All Cluster Nodes are Unconfigured

When all NNMi High Availability (HA) cluster nodes have been unconfigured, the `ov.conf` file no longer contains any mount point references to the NNMi shared disk.

To re-create the mount point reference without overwriting the data on the shared disk, follow these steps on the primary node:

1. If NNMi is running, stop it:


```
ovstop -c
```
2. Reset the reference to the shared disk:
 - *Windows:*

```
%NnmInstallDir%\misc\nnm\ha\nnmhadisk.ovpl NNM -setmount <HA_mount_point>
```
 - *Linux:*

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

3. In the `ov.conf` file, verify the entries related to HA mount points.

For the location of the `ov.conf` file, see ["NNMi High Availability Configuration Files" on page 193](#).

NNMi Does Not Start Correctly Under High Availability

When NNMi does not start correctly, it is necessary to debug 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 debug process, put the system in maintenance mode *without* the `NORESTART` keyword.

1. On the active node in the HA cluster, disable HA resource group monitoring by creating the following maintenance file:

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

2. Start NNMi:

ovstart

3. Verify that NNMi started correctly:

ovstatus -c

All NNMi services should 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\\maintenance`
- *Linux:* `$NnmDataDir/hacluster/<resource_group>/maintenance`

Changes to NNMi Data are Not Seen after Failover

The NNMi configuration points to a different system than where NNMi is running. To fix the problem, verify that the `ov.conf` file has appropriate entries for the following items:

- `NNM_INTERFACE=<virtual_hostname>`
- `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_EVENTDB_DIR=<HA_mount_point>/NNM/dataDir/shared/nnm/eventdb`
- `HA_CUSTOMPOLLER_DIR=<HA_mount_point>/NNM/dataDir/shared/nnm/databases/custompoller`
- `HA_NNM_LOG_DIR=<HA_mount_point>/NNM/dataDir/log`
- `HA_JBOSS_DATA_DIR=<HA_mount_point>/NNM/dataDir/nmsas/NNM/data`
- `HA_LOCALE=C`

For the location of the `ov.conf` file, see ["NNMi High Availability Configuration Files" on page 193](#).

nmsdbmgr Does Not Start after High Availability Configuration

This situation usually occurs as a result of starting NNMi after running the `nmhaconfigure.ovpl` command but without the `nmhadisk.ovpl` command with the `-to` option having been run. In this case, the `HA_POSTGRES_DIR` entry in the `ov.conf` file specifies the location of the embedded database on the shared disk, but this location is not available to NNMi.

To fix this problem, follow these steps:

1. On the active node in the High Availability (HA) cluster, disable HA resource group monitoring by creating the following maintenance file:

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

2. Copy the NNMi database to the shared disk:

- *Windows:*
`%NnmInstallDir%\misc\nnm\ha\nmhadisk.ovpl NNM
-to <HA_mount_point>`
- *Linux:*
`$NnmInstallDir/misc/nnm/ha/nmhadisk.ovpl NNM
-to <HA_mount_point>`

Caution: To prevent database corruption, run this command (with the `-to` option) only one time. For information about alternatives, see ["Re-Enable NNMi for High Availability after All Cluster Nodes are Unconfigured" on page 187](#).

- *Windows:*
`%NnmInstallDir%\misc\nnm\ha\nmhasstartrg.ovpl NNM <resource_group>`
- *Linux:*
`$NnmInstallDir/misc/nnm/ha/nmhasstartrg.ovpl NNM <resource_group>`

3. Start NNMi:

`ovstart`

4. Verify that NNMi started correctly:

`ovstatus -c`

All NNMi services should show the state RUNNING.

5. After completing your troubleshooting, delete the maintenance file:

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

NNMi Runs Correctly on Only One High Availability Cluster Node (Windows)

The Windows operating system requires two different virtual IP addresses, one for the High Availability (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 only runs correctly 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 for the network.

Disk Failover Does Not Occur

This situation can happen 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 all work together.

If 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.

Shared Disk is Not Accessible (Windows)

The command `nmhaclusterinfo.ovpl -config NNM -get HA_MOUNT_POINT` returns nothing.

The drive of the shared disk mount point must be fully specified (for example, `S:\`) during HA configuration.

To correct this problem, run the `nmhaconfigure.ovpl` command on each node in the HA cluster. Fully specify the drive of the shared disk mount point.

Shared Disk Does Not Contain Current Data

Responding to the `nmhaconfigure.ovpl` command question about disk type with the text `none` bypasses the code for setting the disk-related variables in the `ov.conf` file. To fix this situation, follow the procedure in ["Prepare the Shared Disk Manually in High Availability Environments" on page 174](#).

Shared Disk Files Are Not Found by the Secondary Node after Failover

The most common cause of this situation is that the `nmhadisk.ovpl` command was run with the `-to` option when the shared disk was not mounted. In this case, the data files are copied to the local disk, so the files are not available on the shared disk.

To fix this problem, follow these steps:

1. On the active node in the High Availability (HA) cluster, disable HA resource group monitoring by creating the following maintenance file:
 - *Windows:* `%NnmDataDir%\hacluster\\maintenance`
 - *Linux:* `$NnmDataDir/hacluster/<resource_group>/maintenance`
2. Log on to the active node, and then verify that the disk is mounted and available.
3. Stop NNMi:

ovstop

- 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>
```

Caution: To prevent database corruption, run this command (with the `-to` option) only one time. For information about alternatives, see ["Re-Enable NNMi for High Availability after All Cluster Nodes are Unconfigured" on page 187](#).

- 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>
```

- Start NNMi:

ovstart

- Verify that NNMi started correctly:

ovstatus -c

All NNMi services should show the state RUNNING.

- After completing your troubleshooting, delete the maintenance file:

- Windows:* %NnmDataDir%\hacluster*<resource_group>*\maintenance

- Linux:* \$NnmDataDir/hacluster/*<resource_group>*/maintenance

Error: Wrong Number of Arguments

The name of the product Perl module is a required parameter to most of the NNMi High Availability (HA) configuration commands.

- For NNMi, use the value NNM.
- To determine what value to use for an NNM iSPI, see the documentation for that NNM iSPI.

Resource Hosting Subsystem Process Stops Unexpectedly (Windows Server)

Starting an High Availability (HA) cluster resource on a computer running the Windows Server operating system stops the Resource Hosting Subsystem (Rhs.exe) process unexpectedly.

For information about this known problem, see the Microsoft Support web site article *The Resource Hosting Subsystem (Rhs.exe) process stops unexpectedly when you start a cluster resource in Windows Server*, which is available from <http://support.microsoft.com/kb/978527>.

Tip: Always run the NNMi resource in a separate resource monitor (rhs.exe) specific to the resource group.

Product Startup Times Out (Windows WSCS 2008)

After upgrading to NNMi 10.30, if the app resource (<resource>-app) in the Failover Cluster Manager changes from "Pending" to "Failed", there might be a timeout issue. If this situation occurs, do the following:

1. Use the `cluster log /gen` command to generate the `cluster.log` file.
2. Open the log located in the following directory:

```
C:\Windows\cluster\reports\cluster.log
```

3. If you see an error in the `cluster.log` file similar to the following, you have a `DeadlockTimeout` issue:
 ERR [RHS] Resource <resource-name>-APP handling deadlock. Cleaning current operation.
 The `DeadlockTimeout` is the total time for failover when the agent might be blocked. The `PendingTimeout` represents either the online or offline operation. The `DeadlockTimeout` default value is 45 minutes (2,700,000 milliseconds), and the `PendingTimeout` default value is 30 minutes (1,800,000 milliseconds).

You can change the `DeadlockTimeout` and the `PendingTimeout` values. For example, to set a `DeadlockTimeout` of 75 minutes and a `PendingTimeout` of 60 minutes, you can run the following commands:

```
cluster res "<resource group>-APP" /prop DeadlockTimeout=4500000
cluster res "<resource group>-APP" /prop PendingTimeout=3600000
```

See your High Availability vendor documentation for more information

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.

Cannot Start the NNMi HA Resource Group on a Particular Cluster Node

If the `nmhastartrg.ovpl` or `nmhastartrg.ovpl` command does not correctly start, stop, or switch the NNMi HA resource group, review the following information:

- **MSFC:**
 - In Failover Cluster Management, review the state of the NNMi HA resource group and underlying resources.
 - Review the Event Viewer log for any errors.

- VCS:
 - Run `/opt/VRTSvcs/bin/hares -state` to review the resource state.
 - For failed resources, review the `/var/VRTSvcs/log/<resource>.log` file for the resource that is failing. Resources are referenced by the agent type, for example: `IP*.log`, `Mount*.log`, and `Volume*.log`.

If you cannot locate the source of the problem, you can manually start the NNMi HA resource group by using the HA product commands:

1. Mount the shared disk.
2. Assign the virtual host to the network interface:
 - *MSF*:
 - Start Failover Cluster Management.
 - Expand the resource group.
 - Right-click `<resource_group>-ip`, and then click **Bring Online**.
 - *VCS*: `/opt/VRTSvcs/bin/hares -online <resource_group>-ip -sys <Local_hostname>`
 - *RHCS*: Run `/usr/sbin/cmmmodnet` to add the IP address.
3. Start the NNMi HA resource group. For example:
 - *Windows*:


```
%NnmInstallDir%\misc\nnm\ha\nnmhastartrg.ovpl NNM
-start <resource_group>
```
 - *Linux*:


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

The return code 0 indicates that NNMi started successfully.

The return code 1 indicates that NNMi did not start correctly.

High Availability Configuration Reference

This section contains reference information for the following High Availability configuration items:

["NNMi High Availability Configuration Files" below](#)

["NNMi-Provided HA Configuration Scripts" on the next page](#)

["NNMi High Availability Configuration Log Files" on page 195](#)

NNMi High Availability Configuration Files

The following table lists the NNMi High Availability (HA) configuration files. These files apply to NNMi and add-on NNM iSPIs on the NNMi management server. These files are installed to the following location:

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

NNMi HA Configuration Files

File Name	Description
ov.conf	Updated by the <code>nnmhaclusterinfo.ovpl</code> command to describe the NNMi HA implementation. NNMi processes read this file to determine the HA configuration.
nnmdatareplicator.conf	Used by the <code>nnmdatareplicator.ovpl</code> command to determine which NNMi folders and files are included in data replication from the active node to the passive nodes. If you implement a different method of replicating the NNMi configuration, see this file for a list of the data to include. For more information, see the comments in the file.

NNMi-Provided HA Configuration Scripts

The following tables list the HA configuration scripts that are included with NNMi. The NNMi-provided scripts listed in [NNMi HA Configuration Scripts](#) are convenience scripts that can be used to configure HA for any product that has a customer Perl module. If you prefer, you can use the HA product-provided commands to configure HA for NNMi.

On the NNMi management server, the NNMi-provided HA configuration scripts are installed to the following location:

- *Windows:* %NnmInstallDir%\misc\nnm\ha
- *Linux:* \$NnmInstallDir/misc/nnm/ha

NNMi HA Configuration Scripts

Script Name	Description
nnmhaconfigure.ovpl	Configures NNMi or an NNM iSPI for an HA cluster. Run this script on all nodes in the HA cluster.
nnmhaunconfigure.ovpl	Unconfigures NNMi or an NNM iSPI from an HA cluster. Optionally, run this script on one or more nodes in the HA cluster.
nnmhaclusterinfo.ovpl	Retrieves cluster information regarding NNMi. Run this script as needed on any node in the HA cluster.
nnmhadisk.ovpl	Copies NNMi and NNM iSPI data files to and from the shared disk. During HA configuration, run this script on the primary node. At other times, run this script per the instructions in this chapter.
nnmhastartrg.ovpl	Starts the NNMi HA resource group in an HA cluster. During HA configuration, run this script on the primary node.
nnmhastoprg.ovpl	Stops the NNMi HA resource group in an HA cluster. During HA unconfiguration, run this script on the primary node.

The NNMi-provided scripts listed in the following table are used by the scripts listed in [NNMi HA Configuration Scripts](#). Do not run the scripts listed in the following table directly.

NNMi HA Support Scripts

Script Name	Description
nmmdatareplicator.ovpl	Checks the nmmdatareplicator.conf configuration file for changes and copies files to remote systems.
nmmharg.ovpl	Starts, stops, and monitors NNMi in an HA cluster. For VCS configurations, used by the VCS start, stop, and monitor scripts. (nmmhargconfigure.ovpl configures this usage.) Also used by nmhastarttrg.ovpl to enable and disable tracing.
nmmhargconfigure.ovpl	Configures HA resources and resource groups. Used by nmhmaconfigure.ovpl and nmhmaunconfigure.ovpl.
nmhastart.ovpl	Starts NNMi in an HA cluster. Used by nmmharg.ovpl.
nmhastop.ovpl	Stops NNMi in an HA cluster. Used by nmmharg.ovpl.
nmhamonitor.ovpl	Monitors NNMi processes in an HA cluster. Used by nmmharg.ovpl.
nmhamsocs.vbs	Is a template for creating a script to start, stop, and monitor NNMi processes in a MSFC HA cluster. The generated script is used by MSFC and is stored in the following location: %NnmDataDir%\hacluster\ <i><resource_group></i> \hamsocs.vbs

NNMi High Availability Configuration Log Files

The following log files apply to the HA configuration for NNMi and add-on NNM iSPIs on the NNMi management server:

- *Windows* configuration:
 - %NnmDataDir%\tmp\HA_nnmhaserver.log
 - %NnmDataDir%\log\haconfigure.log
- *Linux* configuration:
 - \$NnmDataDir/tmp/HA_nnmhaserver.log
 - \$NnmDataDir/log/haconfigure.log
- *Windows* runtime:
 - Event Viewer log
 - %HA_MOUNT_POINT%\NNM\dataDir\log\nnm\ovspmd.log
 - %HA_MOUNT_POINT%\NNM\dataDir\log\nnm\public\postgres.log
 - %HA_MOUNT_POINT%\NNM\dataDir\log\nnm\public\nmsdbmgr.log

- %HA_MOUNT_POINT%\NNM\dataDir\log\nnm\nnm.log
- %SystemRoot%\Cluster\cluster.log
This is the log file for cluster runtime issues including: adding and removing resources and resource groups; other configuration issues; starting and stopping issues.
- *Linux*:
 - /var/adm/syslog/syslog.log
 - \$HA_MOUNT_POINT/NNM/dataDir/log/nnm/ovspmd.log
 - \$HA_MOUNT_POINT/NNM/dataDir/log/nnm/public/postgres.log
 - \$HA_MOUNT_POINT/NNM/dataDir/log/nnm/public/nmsdbmgr.log
 - \$HA_MOUNT_POINT/NNM/dataDir/log/nnm/nnm.log

Tip: You might also need to consult your HA vendor logs. For example, Veritas stores log files in the /var/VRTSvcS/log folder. RHCS records log messages to syslog.

Chapter 5: Modify Default Settings

After you have your NNMi management server functioning, you can optimize several of the NNMi features by modifying default settings.

Modify Access Control Lists for NNMi Folders

You might run across situations that would cause you to modify the user name that runs the HPE NNM Action Server as shown in ["Set the Action Server Name Parameter" on page 200](#). If you change the user name that runs the action server without modifying the user name permissions, the HPE NNM Action Server might not start, and NNMi might not log messages when running incident actions. This section includes actions to take to prevent this from happening.

NNMi (Everest) contains permission changes to the following directories:

- `/var/opt/OV/log/nnm/public`
- `/var/opt/OV/shared/perfSpi`

Although the NNMi Everest out-of-the-box permissions for the `/var/opt/OV/log/nnm/public` folder is 755, NNMi uses ACLs to adjust access permissions for the database user (`nmsdbmgr`), and the `nnmaction` user (`bin`). During the NNMi Everest 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 on 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 and the NNMi installation should complete 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 administration documentation for setting 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 to extend the owner/group/other permissions. ACLs are supported on all the following Linux platforms: RedHat and SuSE.

For example, after running the following command, the user depicted by the `USER` variable obtains write access to the `/var/opt/OV/log/nnm/public` folder. 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 information about how to use the `setfacl` command, see the Linux manpage.

Change the Custom Poller Collections Export Directory

NNMi writes the data from the collections you export into the following directory:

- *Windows*: %NnmDataDir%\shared\nnm\databases\custompoller\export
- *Linux*: \$NnmDataDir/shared/nnm/databases/custompoller/export

To change the directory that NNMI writes its custom poller files into, 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, which 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
```

3. Restart the NNMI management server.

- Run the `ovstop` command on the NNMI management server.
- Run the `ovstart` command on the NNMI management server.

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:

a. Edit the following file:

- *Windows*: %NNM_PROPS%\nms-custompoller.properties
- *Linux*: \$NNM_PROPS/nms-custompoller.properties

b. Look for the `maxdiskspace` entry, which is similar to the following line:

```
#!com.hp.nnm.custompoller.maxdiskspace=1000
```

To configure NNMI to reserve up to 2000 MB (2 GB) of storage space for each `collection_name.csv` file, change the line as follows:

```
com.hp.nnm.custompoller.maxdiskspace=2000
```

c. Restart the NNMI management server.

- Run the `ovstop` command on the NNMI management server.
- Run the `ovstart` command on the NNMI management server.

Configure Incident Actions

You can configure actions to automatically run at any point in the incident lifecycle. For example, you might want to configure an action to occur when an incident of the type you are configuring is generated. See *Configure an Action for an Incident* in the NNMI Help for more information.

To adjust action parameters, follow the steps shown in the following sections.

Note: To avoid undesirable results (such as unintended memory growth, slower event action processing time), HPE recommends that you do not change the default property values for event action processing.

Set 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%\shared\nnmaction.properties
 - *Linux:* \$NNM_PROPS/shared/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
```

Note: Make sure to remove the **#!** characters located at the beginning of the line.

3. Restart the NNMi management server.
 - a. Run the **ovstop** command on the NNMi management server.
 - b. Run the **ovstart** command on the NNMi management server.

Set 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%\shared\nnmaction.properties
 - *Linux:* \$NNM_PROPS/shared/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
```

Note: Make sure to remove the **#!** characters located at the beginning of the line.

3. Restart the NNMi management server.
 - a. Run the **ovstop** command on the NNMi management server.
 - b. Run the **ovstart** command on the NNMi management server.

Set the Action Server Name Parameter

If you have an NNMi management server running on a Windows operating system, the HPE NNM Action Server runs as a windows service with a Local System account. That means you must use the Local System account to run action server actions.

To modify the user name that runs the HPE NNM Action Server windows service on a Windows NNMi management server, change the LogOn property of the HPE NNM Action Server service.

If you have an NNMi management server running on a Linux operating system, the action server runs with a bin user name. To modify the user name that runs the action server on these operating systems, complete the following steps:

1. Edit the following file:

```
$NNM_PROPS/nnmaction.properties
```

2. Look for a line the resembles the following:

```
#!com.hp.ov.nms.events.action.userName=bin
```

To configure NNMi to have *root* run the action server instead of the default value, change the line as follows:

```
com.hp.ov.nms.events.action.userName=root
```

Note: Make sure to remove the **#!** characters located at the beginning of the line.

3. Save your changes.
4. Restart the action server:
 - a. Run the `ovstop nnmaction` command on the NNMi management server.
 - b. Run the `ovstart nnmaction` command on the NNMi management server.

Change the Action Server Queue Size

For actions that use a long action command string at a high execution rate, such as responding to a trap storm, the action server can use up a lot of memory. To provide better action server performance, HPE places limits on the memory size to which the action server can grow.

To modify these limits, follow these steps:

1. Edit the following file:
 - `%NNM_PROPS%\shared\nnmaction.properties`
 - `$NNM_PROPS/shared/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. The above parameters show the minimum memory size set to 6MB and the maximum set to 30MB. Adjust these parameters to meet your needs.
4. Save your changes.

5. Restart the NNMI management server.
 - a. Run the **ovstop** command on the NNMI management server.
 - b. Run the **ovstart** command on the NNMI management server.

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, use the **Tools > Incident Actions Log** menu option. The following table describes the items contained in the log:

Incident Actions Log Items

Item	Description
Command	Script to run when incident occurs
Incident Name	Name of incident as defined in incident configuration
Incident UUID	The UUID of the incident (from Registration tab)
Command Type	Type of command (Jython or ScriptOrExecutable)
Lifecycle State	Lifecycle state of the incident (Registered , In Process , Completed , or Closed)
Exit Code	Return code of the command (similar to an error code)
Standard Output	Standard output of the action
Standard Error	Standard error output
Execution Status	The determined status per the action

Overriding Settings in the server.properties File

Note: Note that a system might have two server.properties files.

The following file is created by the product installer and contains properties that customize the application server for the application instance. This file is *not* modified by customers and is replaced during code maintenance (upgrades and patches).

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

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

The following file is used by customers to configure the application for their environment and will not be modified by the product during upgrades or patches. This file overrides values configured in other files. So all customizing is done in this file.

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

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

This section describes how to override the following settings in the nmsas/NNM/server.properties file:

Override the Browser Locale Setting

You can use the following `server.properties` file to force the given Locale value for all NNMi clients regardless of the browser Locale value:

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

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

When this value is set using the `server.properties` file, the browser Locale value is ignored.

To override the browser Locale setting:

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, to set the Locale to English using only the ISO Language code, enter the following:

`nmsas.server.forceClientLocale = en`

`nmsas.server.forceClientLocale= <two-letter ISO Language code>_<two-letter ISO country code>`

For example, to set the Locale to English using the ISO Language and Country codes, enter the following:

`nmsas.server.forceClientLocale = en_US`

4. Restart the NNMi `ovjboss` service:

Run the `ovstop ovjboss` command on the NNMi management server.

Run the `ovstart` command on the NNMi management server.

Note: Changes to the `server.properties` file are read only at `ovjboss` startup.

See comments in the `server.properties` file for more information.

Configure the Locale for Sort Order of User Names when Assigning Incidents

An NNMi administrator can specify the language locale for the NNMi management server that should be used to determine the sort order of user names when assigning incidents.

Note: The configured sort order locale is applied only to the **Assign Incidents** dialog.

When determining alphabetical order, NNMi uses the user display name rather than the actual login name and does not sort capital letters separately from lowercase.

Note: NNMi uses only the locale configured in `sortLocale` for determining sort order. The browser locale specified in the `forceClientLocale` property does not affect the sort order. For more information, see ["Override the Browser Locale Setting" above](#)

Note: When making file changes under High Availability (HA), the location of the `server.properties` file that you need to update is: `<Shared_Disk>/NNM/dataDir/nmsas/NNM/server.properties`.

To configure the language locale to use for the sort order for the user names listed when you assign an incident, edit the `server.properties` file as follows:

1. Open the following file:
 - *Windows:* `%NnmDataDir%\nmsas\NNM\server.properties`
 - *Linux:* `$NnmDataDir/nmsas/NNM/server.properties`
2. Uncomment the following line in the `server.properties` file:
`#nmsas.server.sortLocale = en_US`
3. Change the default value to the correct locale for your NNMi management server. For example, to change the locale to the Russian language, use the following entry:
`nmsas.server.sortLocale = ru_RU`
4. Restart the NNMi management server.
 - a. Run the `ovstop` command on the NNMi management server.
 - b. Run the `ovstart` command on the NNMi management server.

Configure SNMP Set Object Access Privilege

You can use the following file to configure the Object Access Privilege 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`

See the NNMi Help for Operators for more information about the SNMP Set feature. See the NNMi Help for Administrators for more information about Object Access Privileges.

To configure the Object Access Privilege 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>`
Valid values for `<object access role>` include the following:
`com.hp.nnm.ADMIN`
`com.hp.nnm.LEVEL2`
`com.hp.nnm.LEVEL1`
`com.hp.nnm.GUEST`

For example, to enable **Object Administrator** and **Object Operator Level 2** Object Access Privileges to use the SNMP Set feature, type the following:

```
permission.override.com.hp.nnm.SNMP_SET=com.hp.nnm.ADMIN,com.hp.nnm.LEVEL2
```

3. Include each Object Access Privilege for which you want to enable access.
4. Restart the NNMi ovjboss service:
 - Run the `ovstop ovjboss` command on the NNMi management server.
 - Run the `ovstart` command on the NNMi management server.

Note: Changes to the `server.properties` file are only read at ovjboss startup.

Configure NNMi to Require Encryption for Remote Access

An administrator can disable HTTP and other unencrypted access from the network to NNMi.

Note: Before configuring NNMi to permit only encrypted remote access, make sure Global Network Management, NNM iSPIs, 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 and other unencrypted access from the network to NNMi, edit the `server.properties` file as follows:

1. Edit the following file (you may 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. Restart the NNMi management server.
 - a. Run the `ovstop` command on the NNMi management server.
 - b. Run the `ovstart` command on the NNMi management server.

With the modification just described, NNMi will not “listen” to HTTP requests from a remote system; however, HTTP requests would still be supported for localhost access.

Modify User Interface Properties

This section describes how to set the following user interface properties in the `ui.properties` file:

Modify NNMi Gauge Titles to Show SNMP MIB Variable Names

The **Node Sensor Gauges** and **Physical Sensor Gauges** tabs in the NNMi analysis pane contains gauges showing the NNMi component name as the MIB OID being polled. This helps you understand which gauge goes with which component. The Node Sensor name helps differentiate gauges if NNMi shows many gauges for a node. For example, if a node contains a large number of CPUs, NNMi shows different names for the individual CPUs.

With this feature disabled, NNMi shows the SNMP MIB variable name to be the same for all CPUs.

If you want to change this property to show gauge titles as SNMP MIB variable names rather than NNMi Node Sensor names, complete the following 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:


```
com.hp.nnm.ui.analysisGaugeTitleIsNodeComponentName = true
```
3. Edit the following line to read as follows:


```
com.hp.nnm.ui.analysisGaugeTitleIsNodeComponentName = false
```
4. Save your changes.
5. Restart NNMi:
 - a. Run the **ovstart** command on the NNMi management server.
 - b. Run the **ovstop** command on the NNMi management server.

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. See "[Maintenance Mode](#)" on page 177 for more information.

Modify 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.

Note: 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:
 - a. Run the **ovstop** command on the NNMi management server.
 - b. Run the **ovstart** command on the NNMi management server.

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. See "[Maintenance Mode](#)" on page 177 for more information.

Enable 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.

Caution: 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. Open 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.LEVEL2`
3. Save the file.
 4. Restart NNMi.
 - Run the `ovstop` command on the NNMi Management Server.
 - Run the `ovstart` command on the NNMi Management Server.

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. See "[Maintenance Mode](#)" on page 177 for more information.

Enable Level 2 Operators to Edit Node Group Maps

By default, NNMi permits NNMi administrators to edit maps by creating, modifying, and deleting Node Groups. You can configure accounts assigned to the NNMi Operator Level 2 User Group to have this ability as well.

If you must change 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, do the following:

1. Open 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 changes.
4. Restart NNMi:
 - a. Run the `ovstart` command on the NNMi management server.
 - b. Run the `ovstop` command on the NNMi management server.

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. See "[Maintenance Mode](#)" on page 177 for more information.

After completing step 1 through step 4, the NNMiconsole changes as follows:

- The **Inventory > Node Group** menu shows a create and Delete toolbar icon for the NNMi Operator Level 2.
- The **Inventory > Node Group** menu contains an **Action > Delete** menu item for the NNMi Operator Level 2.
- The **All Node Groups** folder appears in the **Topology Maps** workspace. See "About Workspaces" in the NNMi online help for more information.

- For Node Group maps, the NNMi console contains the **Save Layout** toolbar button and **File > Save Layout** menu items.
- The behavior of the **Save Layout** Action menu depends on the whether Node Group Map settings are configured for the Node Group map. If no Node Group Map Setting exists for a Node Group map, you must create one.

You can also configure NNMi so that NNMi Operator Level 2 users have permission to create a Node Group Map Setting:

1. From the NNMi console, open **Topology Maps > Node Group Overview**.
2. Double-click the Node Group 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 can now create, edit and delete Node Group Map Settings from a Node Group Map view.

Enabling Level 1 Operators to Run Status and Configuration Polls

NNMi permits User Accounts assigned to the NNMi Operator Level 2 User Group to run Status Poll and Configuration Poll 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.

To change the Menu Item configuration NNMi to permit User Accounts assigned to the NNMi Operator Level 1 User Group to view the Status Poll menu item, do the following:

1. Open the **Configuration->User Interface->Menu Items->Status Poll** form.
2. From the **Menu Items** tab, scroll to the **Status Poll** menu item label.
3. From the **Menu Item Contexts** tab, open the entry for each **Required NNMi Role** and **Object Type** item you must change.
4. Change the value of the **Required NNMi Role** to Operator Level 1 for each object type you want a Level 1 operator to be able to status poll.

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** and **Object Type** item you must change.
3. Change the value of the **Required NNMi Role** to Operator Level 1 for each object type you want a Level 1 Operator to be able to configuration poll.

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.

Note: You must edit the `nms-topology.properties` file to permit User Accounts assigned to the

NNMiOperator Level 1 User Group to run both the Status Poll and Configuration Poll commands from the NNMi console. If you do not complete these steps, NNMi displays the Status Poll and Configuration Poll options in the Actions menu, but the user views an error message when attempting to run the Status Poll or Configuration Poll commands.

To change the level of access required for the status poll and configuration poll (the required Object Access Privilege levels),

1. Open the following file:

Windows: %NNM_PROPS%\nms-topology.properties

Linux: \$NNM_PROPS/nms-topology.properties

2. Scroll to the bottom of the file, then 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
```

3. 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
```

4. Save your changes.
5. Restart the NNMi management server:
 - a. Run the ovstop command on the NNMi management server.
 - b. Run the ovstart command on the NNMi management server.

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. See "[Maintenance Mode](#)" on page 177 for more information.

Configure the Data Payload Size in an ICMP Echo Request Packet

One definition of network latency is the time for an ICMP packet to complete a round trip to the target device and back. A low latency measurement indicates a more efficient network.

One common way to test network latency is to adjust the ICMP polling frequency and ICMP echo request packet data payload size for a management address being managed by NNMi. Considering that a larger packet has a longer network latency than a smaller one, NNMi permits you to experiment with different packet sizes to measure the network latency.

You can configure the size of the data payload NNMi sends in an ICMP echo request packet for IP addresses that belong to nodes in a node group or interfaces in an interface group. For example, you might modify the size of the ICMP echo request packets sent to node groups or interface groups in conjunction with adjusting management address polling times to compare network latency.

To configure a different payload size for addresses that belong to nodes in a node group and interfaces in an interface group, complete the following steps:

1. Edit the following file:

Windows: %NNM_PROPS%\nms-mon-config.properties

Linux: \$NNM_PROPS/nms-mon-config.properties

2. Locate the text block containing the following:

```
#!com.hp.nnm.icmp.payload.sizeInBytes=4096
```

3. Uncomment and edit the line to read as follows, changing the 4096 value to the payload value you need:

```
com.hp.nnm.icmp.payload.sizeInBytes=4096
```

The minimum value to use for the `sizeInBytes` parameter is 12 bytes and the maximum value is 65492 bytes.

Note: To configure the data payload size at least one of the group properties must be defined. If neither of the group properties are defined as described in the following steps, NNMi ignores the `com.hp.nnm.icmp.payload.sizeInBytes` property.

1. Locate the text block containing the following:

```
#!com.hp.nnm.icmp.nodegroup.name=My Node Group
```

2. Uncomment and edit the line to read as follows, changing the My Node Group setting to the node group you plan to reference by NNMi monitoring settings:

```
com.hp.nnm.icmp.nodegroup.name=My Node Group
```

Note: The node group name you specify needs to be a node group referenced by NNMi monitoring settings.

3. Locate the text block containing the following:

```
#!com.hp.nnm.icmp.ifacegroup.name=My Interface Group
```

4. Uncomment and edit the line to read as follows, changing the My Interface Group setting to the interface group you plan to reference by NNMi monitoring settings:

```
com.hp.nnm.icmp.ifacegroup.name=My Interface Group
```

Note: The interface group name you specify needs to be an interface group referenced by NNMi monitoring settings.

5. Restart the NNMi management server

Run the `ovstop` command on the NNMi management server:

Run the `ovstart` command on the NNMi management server.

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. See ["Maintenance Mode" on page 177](#) for more information.

Configure how NNMi Determines the Host Name for a Device

You can change the `HostNameMatchManagementIP` property to `false` to configure NNMi to use the pre- NNMi 9.0 method of finding a valid host name for a discovered device.

Tip: In most cases, keep the default value of this property, which is `true`. See the `nms-disco.properties` file for detailed information about the `HostNameMatchManagementIP` property.

To change the `HostNameMatchManagementIP` property to `false`, do the following:

1. Edit the following file:
 - Windows: `%NNM_PROPS%\nms-disco.properties`
 - Linux: `$NNM_PROPS/nms-disco.properties`
2. Search for the text block containing the following property:


```
HostNameMatchManagementIP=true
```
3. Change the property value as follows:


```
HostNameMatchManagementIP=false
```
4. Save your work.
5. Restart the NNMi management server:
 - Run the `ovstop` command on the NNMi management server.
 - Run the `ovstart` command on the NNMi management server.

NNMi looks up all available IP addresses on loopback interfaces to find a valid host name for a discovered device.

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. See "[Maintenance Mode](#)" on page 177 for more information.

Configure Character Set Encoding Settings for NNMi

Depending on the locale configured for your NNMi management server, you might need to configure the source encodings NNMi uses 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 and edit the following line to read as follows:

```
com.hp.nnm.sourceEncoding=UTF-8
```

4. Modify the UTF-8 property value shown in step 3 using the instructions and examples shown in the `nms-jboss.properties` file.
5. Save your changes.
6. Restart the NNMi management server:

Run the `ovstop` command on the NNMi management server.

Run the `ovstart` command on the NNMi management server.

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. See ["Maintenance Mode" on page 177](#) for more information.

Configure the Time NNMi Waits for an NNM iSPI Licensing Request

If you notice a slow or non-response from the NNMi console, and have one or more of the NNM iSPIs installed, you might need to adjust the amount of time NNMi waits for a response from an NNM iSPI licensing request.

The default amount of time NNMi waits for a response from an NNM iSPI licensing request is 20 seconds.

To change this default value, complete the following steps:

- a. Open the following file:

Windows: `%NNM_PROPS%\nms-jboss.properties`

Linux: `$NNM_PROPS/nms-jboss.properties`

- b. Locate the text block containing the following:

```
#!com.hp.ov.nms.licensing.EXTENSION_WAIT_TIMEOUT=20
```

- c. Uncomment and modify the line to read as follows:

```
com.hp.ov.nms.licensing.EXTENSION_WAIT_TIMEOUT=<time in seconds>
```

For example, to change the response time to 25 seconds, enter the following:

```
com.hp.ov.nms.licensing.EXTENSION_WAIT_TIMEOUT=25
```

Tip: Adjusting this parameter to the optimum value could take some experimenting. Adjust the parameter to a higher value for slower responding NNM iSPIs, such as an overly busy NNM iSPI running on a slower server.

- d. Restart the NNMi management server

Run the `ovstop` command on the NNMi management server.

Run the `ovstart` command on the NNMi management server

Modify NNMi Normalization Properties

NNMi stores both hostnames 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 hostnames, 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. HPE recommends that you adjust the settings in this section during deployment, but before running the initial discovery.

If you run an initial discovery, then decide to change the normalization properties later, you can run the **`nnmnode rediscover.ovpl -all`** script to initiate a full discovery. See the *`nnmnode rediscover.ovpl`* reference page, or the Linux manpage, for more information.

You can change the following properties:

- Normalize discovered node names to UPPERCASE, LOWERCASE, or OFF.
- Normalize discovered hostnames to UPPERCASE, LOWERCASE, or OFF.

To change normalization properties follow these steps:

- Edit the following file:
 - *Windows*: %NNM_PROPS%\nms-topology.properties
 - *Linux*: \$NNM_PROPS/nms-topology.properties
- To configure NNMi to normalize discovered names, look for a line the resembles the following:

```
#!com.hp.ov.nms.topo.NAME_NORMALIZATION=OFF
```

- Un-comment the property:

```
com.hp.ov.nms.topo.NAME_NORMALIZATION=OFF
```

Note: To un-comment a property, remove the `#!` characters from the beginning of a line.

- Change OFF to LOWERCASE or UPPERCASE.
 - Save your changes.
- To configure NNMi to normalize discovered hostnames, look for a line the resembles the following:

```
#!com.hp.ov.nms.topo.HOSTNAME_NORMALIZATION=OFF
```

- Un-comment the property:

```
com.hp.ov.nms.topo.HOSTNAME_NORMALIZATION=OFF
```

- Change OFF to LOWERCASE or UPPERCASE.
- Save your changes.

- Restart the NNMi management server.
 - Run the **`ovstop`** command on the NNMi management server.
 - Run the **`ovstart`** command on the NNMi management server.

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. See ["Maintenance Mode" on page 177](#) for more information.

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, while there are many vendor-specific protocols like *Cisco Discovery Protocol* (CDP) for Cisco devices.

You can configure NNMi to suppress discovery protocol collections for devices you specify. There are special circumstances that might be remedied by suppressing discovery protocol collections.

Here are some examples:

- *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 could prevent this by configuring NNMi to skip EnDP and LLDP processing on these devices. To do this, add the management address of the devices to the `disco.SkipXdpProcessing` file as shown in ["Suppressing the Use of Discovery Protocol Collections" below](#).

Note: New operating system versions on some Enterasys devices support the `set snmp timefilter break` command. On those Enterasys devices, run the `set snmp timefilter break` command. If you configure the device using this command, you do not need to list the device in the `disco.SkipXdpProcessing` file.

- *Nortel devices*: Many Nortel devices use *SynOptics Network Management Protocol* (SONMP) to discover layer 2 layout and connectivity. Some of these devices use the same MAC address on multiple interfaces, and do 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 as shown in ["Suppressing the Use of Discovery Protocol Collections" below](#).

Suppressing the Use of Discovery Protocol Collections

If you want to suppress this collection, 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 the device IP addresses to the `disco.SkipXdpProcessing` file for all of the devices you want to suppress protocol collection for. Follow the instructions show in the `disco.SkipXdpProcessing` reference page, or the Linux manpage.
3. Restart the NNMi management server.
 - a. Run the `ovstop` command on the NNMi management server.
 - b. Run the `ovstart` command on the NNMi management server.

Note: Suppressing the discovery protocol processing of a node or nodes might cause some inaccuracies in the layer 2 layout of the managed network. HPE is not responsible for these inaccuracies.

Note: The `ovjboss` service reads the `disco.SkipXdpProcessing` file on startup. If you make any changes after starting NNMi, restart NNMi as shown in this step.

Note: If you ran the `setsnmp timefilter break` command on any Enterasys devices, remove the device addresses from the `disco.SkipXdpProcessing` file, then restart NNMi as shown in this step. NNMi displays more accurate layer 2 maps when it uses discovery protocols.

See the `disco.SkipXdpProcessing` reference page, or the Linux manpage, for more information.

Suppressing the Monitoring of IP Addresses on Administrative Down Interfaces

NNMi users commonly configure multiple interfaces with the same IP addresses, where one interface is administratively up and its address is responding to ICMP requests, and the other interface is administratively down and not responding to ICMP requests. In such cases, these administratively down interfaces and their IP addresses should not affect node status.

By default, NNMi suppresses the monitoring of IP addresses on interfaces that are administratively down, thereby preventing node status changes.

You can configure whether the monitoring of IP addresses on administratively down interfaces is performed by doing the following:

1. Open the `nms-disco.properties` file in the following location:
 - Windows: `%NnmDataDir%\shared\nnm\conf\props\nms-disco.properties`
 - Linux: `$NnmDataDir/shared/nnm/conf/props/nms-disco.properties`
2. Look for a section in the file that resembles the following:


```
#!com.hp.ov.nms.disco.suppressMonitoringOfAddressOnAdminDownInterface=true
```
3. You can configure the property as follows:

To suppress monitoring of IP addressees on interfaces that are administratively down, uncomment the line to set the property to true (the default setting). The line should resemble the following:

```
com.hp.ov.nms.disco.suppressMonitoringOfAddressOnAdminDownInterface=true
```

To have NNMi monitor IP addresses on interfaces that are administratively down, uncomment the line and edit the property value as follows:

```
com.hp.ov.nms.disco.suppressMonitoringOfAddressOnAdminDownInterface=false
```

4. Save your changes to the `nms-disco.properties` file.
5. Restart the NNMi management server:
 - Run the `ovstop` command on the NNMi management server.
 - Run the `ovstart` command on the NNMi management server.

Suppressing the Use of VLAN-indexing for Large Switches

One of the methods NNMi uses to learn layer 2 connectivity between and among switch devices in a managed network is to retrieve the `dot1dTpFdbTable` (FDB) from the switches. However, for Cisco switches, NNMi must use a `VLAN-indexing` method to retrieve the entire FDB. If there is a large number of VLANs configured on each device, retrieving the FDB with `VLAN-indexing` might take hours to complete.

Cisco switches are often configured to use the Cisco Discovery Protocol (CDP). CDP is considered to be a superior method for learning Layer 2 connectivity. Large switches located in the in the core of the network might contain many VLANs. These switches typically do not have end nodes connected directly to them. If the switches you want to manage do not have end nodes connected directly to them, you might want to suppress the collection of the FDB on these large switches. NNMi still completes the Layer 2 discovery using data collected from CDP. These large switches are prime candidates for suppression of `VLAN-indexing`. Do not suppress `VLAN-indexing` on smaller switches located at the network's edge (often known as access switches) that have many end nodes attached to them.

You can configure NNMi to suppress `VLAN-indexing`. To do this, the NNMi administrator needs to create and add management addresses or address ranges of the large switches to the `disco.NoVLANIndexing` file as shown in "[Suppressing the Use of VLAN-indexing](#)" below. The `ovjboss` service reads the `disco.NoVLANIndexing` file when it starts. If the NNMi administrator makes changes to the `disco.NoVLANIndexing` file after the `ovjboss` service starts, those changes will not take effect until the next time the `ovjboss` service starts. By default, the `disco.NoVLANIndexing` file does not exist. If the `disco.NoVLANIndexing` does not exist, this feature is disabled and NNMi attempts to use `VLAN-indexing` to collect the entire FDB table on all devices.

Suppressing the Use of VLAN-indexing

If you want to disable this `vlan-indexing`, follow these steps:

Note: Suppressing `vlan-indexing` of a node or nodes might cause some inaccuracies in the layer 2 layout of the managed network. HPE is not responsible for these inaccuracies.

1. Create the following file:
 - *Windows:* `%NnmDataDir%\shared\nnm\conf\disco\disco.NoVLANIndexing`
 - *Linux:* `$NnmDataDir/shared/nnm/conf/disco/disco.NoVLANIndexing`

The `disco.NoVLANIndexing` file is case-sensitive.

2. Add the device IP addresses or address ranges to the `disco.NoVLANIndexing` file for all of the devices you want to disable `vlan-indexing` for. Follow the instructions show in the *disco.NoVLANIndexing* reference page, or the UNIX manpage.
3. Restart the NNMi management server.
 - a. Run the `ovstop` command on the NNMi management server.
 - b. Run the `ovstart` command on the NNMi management server.

Note: The `ovjboss` service reads the `disco.NoVLANIndexing` file on startup.

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. See "[Maintenance Mode](#)" on page 177 for more information.

See the *disco.Disco.NoVLANIndexing* reference page, or the Linux manpage, for more information.

Configuring Sensor Status

NNMi includes the following physical sensors and node sensors, which can be monitored to help determine status:

Physical Sensors and Node Sensors

Physical Sensors	Propagates Status to Physical Component by Default?	Node Sensors	Propagates Status to Node by Default?
FAN	Yes	CPU	No
POWER_SUPPLY	Yes	MEMORY	Yes
TEMPERATURE	No	BUFFERS	No
VOLTAGE	No	DISK_SPACE	No
BACK_PLANE	Yes		

Note: 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, in this case, viewing the status of a chassis would be alerted to the fact that a component of that chassis has some kind of failure.

Configuring Physical Sensor Status

You can configure whether a physical sensor propagates its status to the physical component (for example, chassis) level by following the steps in the following sections.

Propagating Physical Sensor Status to a Physical Component

1. If not already present, create a new properties file with the name `nm-apa.properties` in the following directory:

Windows: `%NmDataDir%\shared\nnm\conf\props`

Linux: `$NmDataDir/shared/nnm/conf/props`

2. Within the properties file, use a text editor to include the following text:

```
com.hp.ov.nms.apa.PhysSensorPropagateToPhysicalComponentStatus_<Type>=true
```

where `<Type>` is a Physical Sensor. See ["Configuring Sensor Status" on the previous page](#) for more information.

3. Save the properties file.

4. Restart the NNMI management server:

Run the `ovstop` command on the NNMI management server

Run the `ovstart` command on the NNMI management server.

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. See ["Maintenance Mode" on page 177](#) for more information.

Configuring Physical Sensor Status to not Propagate to the Physical Component

1. If not already present, create a new properties file with the name `nm-apa.properties` in the following directory:

Windows: `%NmDataDir%\shared\nnm\conf\props`

Linux: `$NmDataDir/shared/nnm/conf/props`

2. Within the properties file, use a text editor to include the following text:

```
com.hp.ov.nms.apa.PhysSensorNoPropagateToPhysicalComponentStatus_<Type>=true
```

where `<Type>` is a Physical Sensor. See ["Configuring Sensor Status" on the previous page](#) for more information.

3. Save the properties file.

4. Restart the NNMI management server:

Run the `ovstop` command on the NNMI management server.

Run the `ovstart` command on the NNMI management server.

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. See ["Maintenance Mode" on page 177](#) for more information.

Overriding Physical Sensor Status Values

By default, three sensor state values (None, Warning, and Unavailable) map up to a Normal status by the Causal Engine. You can override these default state mappings such 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. Within the properties file, use a text editor to include one, two, or all three of the following lines, as applicable:

```
com.hp.ov.nms.apa.PhysSensorValueReMappedToDown_NONE=true
```

```
com.hp.ov.nms.apa.PhysSensorValueReMappedToDown_Warning=true
```

```
com.hp.ov.nms.apa.PhysSensorValueReMappedToDown_Unavailable= true
```

3. Save the properties file.
4. Restart the NNMi management server
 - Run the `ovstop` command on the NNMi management server.
 - Run the `ovstart` command on the NNMi management server.

Note: You can map an Unavailable state to an Unpolled status (since Unavailable means that the measurement facility is not available). This situation can often occur because the sensor is non-functional as opposed to the component being non-functional. To map Unavailable to Unpolled, use the same procedure as just described, except in step 2, use the following text:

```
com.hp.ov.nms.apa.PhysSensorValueReMappedToUnpolled_Unavailable= true
```

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. See "[Maintenance Mode](#)" on page 177 for more information.

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 sections.

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. Within the properties file, use a text editor to include the following text:
`com.hp.ov.nms.apa.NodeSensorPropagateToNodeStatus_<Type>=true`
 where <Type> is a Node Sensor. See ["Configuring Sensor Status" on page 217](#) for more information.
3. Save the properties file.
4. Restart the NNMI management server:
 Run the `ovstop` command on the NNMI management server.
 Run the `ovstart` command on the NNMI management server.

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. See ["Maintenance Mode" on page 177](#) for more information.

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. Within the properties file, use a text editor to include the following text:
`com.hp.ov.nms.apa.NodeSensorNoPropagateToNodeStatus_<Type>=true`
 where <Type> is a Node Sensor. See ["Configuring Sensor Status" on page 217](#) for more information.
3. Save the properties file.
4. Restart the NNMI management server:
 Run the `ovstop` command on the NNMI management server
 Run the `ovstart` command on the NNMI management server.

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. See ["Maintenance Mode" on page 177](#) for more information.

Overriding Node Component Status Values

By default, three node component state values (None, Warning, and Unavailable) map up to a Normal status by the Causal Engine. You can override these default state mappings such that None, Warning, and Unavailable map to Critical.

To override node component 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. Within the properties file, use a text editor to include one, two, or all three of the following lines, as applicable:

```
com.hp.ov.nms.apa.NodeComponentValueRemappedToDown_NONE=true
```

```
com.hp.ov.nms.apa.NodeComponentValueRemappedToDown_Warning=true
```

```
com.hp.ov.nms.apa.NodeComponentValueRemappedToDown_Unavailable=true
```

Note: You can map an Unavailable state to an Unpolled status (since Unavailable means that the measurement facility is not available). This situation can often occur because the sensor is non-functional as opposed to the component being non-functional. To map Unavailable to Unpolled, use the following text:

```
com.hp.ov.nms.apa.NodeComponentValueRemappedToUnpolled_Unavailable: true
```

3. Save the properties file.
4. Restart the NNMi management server:

Run the `ovstop` command on the NNMi management server

Run the `ovstart` command on the NNMi management server.

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. See "[Maintenance Mode](#)" on page 177 for more information.

Chapter 6: Maintaining NNMi

This section contains the following chapters:

- ["NNMi Backup and Restore Tools" below](#)
- ["NNMi Logging" on page 230](#)
- ["Changing the Management Server" on page 232](#)

NNMi Backup and Restore Tools

A good backup and restore strategy is key to ensuring the uninterrupted operations of any business. HPE Network Node Manager i Software (NNMi) is an important asset for network operations and should be backed up regularly.

The two types of critical data related to an NNMi installation are as follows:

- Files in the file system
- Data in the relational database (embedded or external)

This chapter explains the tools that NNMi provides for backing up and restoring important NNMi files and data.

This section contains the following topics:

- ["Backup and Restore Commands" below](#)
- ["Backing up NNMi Data" on the next page](#)
- ["Restoring NNMi Data" on page 226](#)
- ["Backup and Restore Strategies" on page 228](#)
- ["Backing up and Restoring the Embedded Database Only" on page 229](#)
- ["Using Backup and Restore Tools in a High Availability \(HA\) Environment" on page 230](#)

Backup and Restore Commands

NNMi provides the following scripts for backing up and restoring NNMi data:

- `nnmbackup.ovp1`—Backs up all necessary file system data (including configuration information) and any data stored in the NNMi embedded database.
- `nnmrestore.ovp1`—Restores a backup that was created by using the `nnmbackup.ovp1` script.
- `nnmbackupembdb.ovp1`—Creates a complete backup of the NNMi embedded database (but not the file system data) while NNMi is running.
- `nnmrestoreembdb.ovp1`—Restores a backup that was created by using the `nnmbackupembdb.ovp1` script.
- `nnmresetembdb.ovp1`—Drops the NNMi embedded database tables. Run the `ovstart` command to recreated the tables.

For command syntax, see the appropriate reference page, or the Linux manpage.

Backing up NNMi Data

The NNMi backup command (`nmbackup.ovp1`) copies key NNMi file system data and some or all of the tables in the NNMi Postgres database to the specified target directory.

Each backup operation stores files in a parent directory called `nm-bak-<TIMESTAMP>` inside the target directory. You can specify a `-noTimestamp` option to save disk space. If you use the `-noTimestamp` option, the parent directory is simply named `nm-bak`. When a backup is performed after a previous backup using the `-noTimestamp` option, the previous backup is renamed `nm-bak.previous`, thereby creating a rolling backup. This renaming is done after the second backup is completed to protect against any loss of backup data.

The NNMi backup command can create a tar archive of the backup data, or you can compress the backup files using your own tools. You can then use any appropriate tool to save a copy of the backup.

Tip: If your NNMi implementation uses Oracle for the main NNMi database, the NNMi backup and restore commands work with the NNMi file system data only. External database maintenance should be handled as part of the existing database backup and restore procedures.

The back up and restore data might or might not include data from any NNM iSPIs installed in your network environment. Check the documentation that came with each NNM iSPI for details.

Caution: Any software that locks files (for example, anti-virus or system backup software), can interrupt NNMi access to the NNMi database. This can cause problems such as an inability to read from or write to a file that is being used by another process, such as an anti-virus application. For the NNMi Postgres database, configure these applications to exclude the NNMi database directory (`%NNM_DB%` on Windows, and `$NNM_DB` on Linux). Use `nmbackup.ovp1` to back up the NNMi database regularly.

See the `nmbackup.ovp1` reference page, or the Linux manpage, for more information.

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 "[Backup Scope](#)" below. For the embedded NNMi database, the `nmsdbmgr` service must be running. For an external database, the backup includes NNMi file system data. NNMi processes do not have to be running to back up an external database.
- 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 embedded NNMi database, the backup copies the Postgres database files. For an external database, the backup includes NNMi file system data only.

Backup Scope

The NNMi backup command provides several scopes that define how much NNMi is backed up.

Configuration scope

The configuration scope (-scope config) loosely aligns to the information in the **Configuration** workspace of the NNMi console.

The configuration scope includes the following data:

- For online backups, only those embedded database tables that store NNMi configuration information.
- For offline backups, the entire embedded database.
- For all backups, the NNMi configuration information in the file system as listed in [Configuration Scope Files and Directories](#).

Topology scope

The topology scope (-scope topology) loosely aligns to the information in the **Inventory** workspace of the NNMi console. Because the network topology is dependent on the configuration that was used for discovering that topology, the topology scope includes the configuration scope.

The topology scope includes the following data:

- For online backups, only those embedded database tables that store NNMi configuration and network topology information.
- For offline backups, the entire embedded database.
- For all backups, the NNMi configuration information in the file system as listed in the first of the following tables. 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 of the NNMi console. Because events are dependent on the network topology related to those events, the event scope includes the configuration and topology scopes.

The event scope includes the following data:

- For online backups, only those embedded database tables that store NNMi configuration, network topology, and event information.
- For offline backups, the entire embedded database.
- For all backups, the NNMi configuration information in the file system as listed in the first of the following tables and the NNMi event information as listed in [Event Scope Files and Directories](#).

All scope

The complete backup (-scope all) includes all important NNMi files and the complete embedded database.

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
%NnmInstallDir%\nmsas\server\nms\conf \$NnmInstallDir/nmsas/server/nms/conf	jboss configuration
%NnmDataDir%\conf \$NnmDataDir/conf	Configuration that might be shared by other HPE products

Configuration Scope Files and Directories, continued

Directory or File name	Description
%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-snmplib \$NnmDataDir/shared/nnm/user-snmplib	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
%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 maps

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.

Event Scope Files and Directories

Directory or File name	Description
\$NnmDataDir/log/nnm/signin.0.0.log	NNMi console sign-in log

Restoring NNMi Data

The NNMi restore script (`nnmrestore.ovp1`) places the backup data on the NNMi management server. The type and scope of the backup determines what NNMi can restore.

Note: If you use the `nnmrestore.ovp1` script to place database records on a second NNMi management server, both NNMi management servers must have the same type of operating system and NNMi version and patch level.

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, uninstall NNMi from the original NNMi management server.

Before uninstalling NNMi, remove any NNMi patches in reverse order, beginning with the most recent patch. The patch removal process varies according to the operating system running on the NNMi management server. See the patch documentation for installation and removal instructions.

- To restore 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 have been deleted since the backup are restored, and objects that have been created since the backup are deleted. Additionally, any objects that were changed after the backup was taken revert to their state at the time of the backup. For the embedded NNMi database, the `nmsdbmgr` service must be running. For an external database, the restore includes NNMi file system data only and no NNMi processes must be running.
- To restore an offline backup, NNMi overwrites the Postgres files in the file system, completely replacing the database files with the contents of the backup. For an external database, the backup includes NNMi file system data only.

With the `-force` option, the `nnmrestore.ovp1` command stops all NNMi processes, starts the `nmsdbmgr` service (if restoring from an online backup of the NNMi embedded 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 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.

Note: NNMi automatically resynchronizes topology, state, and status following a restore from backup.

Avoid stopping NNMi during the resynchronization. To help ensure resynchronization has completed, NNMi should remain running for several hours following the restore from backup. The actual time required depends on the number of nodes and the volume of state changes and trap data received while performing the resynchronization.

If NNMi must be stopped before the resynchronization is finished, the resynchronization should be run again and allowed to complete.

To perform a manual resynchronization of the entire management server, run: `nnmnode rediscover.ovp1 -all -fullsync`

Same System Restore

You can use the backup and restore commands on a single system for data recovery. The following items must not have changed between the time of the backup and time of the restore:

- NNMi version (including any patches)
- Operating system type
- Character set (language)
- Hostname
- Domain

Different System Restore

You can 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 system failure and transferring NNMi to a different system during an operating system upgrade.

Note: Because the NNMi UUID is copied to the target system during the database restore, both source and target systems now appear to be running the same instance of NNMi. Uninstall NNMi from the source system.

Before uninstalling NNMi, remove any NNMi patches in reverse order, beginning with the most recent patch. The patch removal process varies according to the operating system running on the NNMi management server. See the patch documentation for installation and removal instructions.

Tip: 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 and version
- Character set (language)

The following items can differ between the two systems:

- Hostname
- Domain

For a different system restore, the `nnmrestore.ovpl` command does not copy the license information to the new system. Obtain and apply a new license for the new NNMi management server. For more information, see ["Apply Licenses" on page 241](#).

Restore on an NNMi Management Server Upgraded to 10.30

If you take a backup on a management server where NNMi 10.30 is newly installed, and then try to restore the backup on a management server where an older version of NNMi is upgraded to the version 10.30, you must perform an additional task before the restore process begins. You must follow the instructions in ["Configure an Upgraded NNMi Environment to Use the New Keystore" on page 245](#).

When you try to perform this type of restore operation without completing the steps in ["Configure an Upgraded NNMi Environment to Use the New Keystore" on page 245](#), the following error messages appear in the `boot.log` file:

```
ERROR [Http11Protocol] Error initializing endpoint: java.io.IOException: FIPS mode:
KeyStore must be from provider JsafeJCE
```

Restore in an HA Cluster

Backup and restore work seamlessly in an NNMi HA cluster.

If you take a backup of the NNMi data on an NNMi management server (standalone or HA cluster) that was upgraded to the version 10.30 from an older version, and then restore the data in a newly installed NNMi 10.30 in an HA cluster, the NNMi processes fail to start after the restore operation is complete.

In this scenario, before taking the backup on the management server that was upgraded to the version 10.30 from an older version, perform additional configuration steps provided in ["Configure an Upgraded NNMi Environment to Use the New Keystore" on page 245](#).

Backup and Restore Strategies

This section discusses the following backup and restore strategies:

- ["Back up All Data Periodically" below](#)
- ["Back up Data Before Changing the Configuration" on the next page](#)
- ["Back up Data Before Upgrading NNMi or the Operating System" on the next page](#)
- ["Restore File System Files Only" on the next page](#)

Back up All Data Periodically

Your disaster recovery plan should include 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, use the `-force` option to ensure that NNMi is on the correct state before the backup begins. For example:

```
nmmbackup.ovpl -force -type online -scope all -archive
  -target nmi_backups\periodic
```

If you must recover your NNMi data after a hardware failure, follow these steps:

1. Rebuild or acquire new hardware.
2. Install NNMi to the same version and patch level as were in place for the backup.
3. Restore the NNMi data:
 - If the recovery NNMi management server meets the requirements listed in ["Same System Restore" on the previous page](#), run a command similar to the following example:


```
nmrestore.ovpl -force -lic
  -source nmi_backups\periodic\newest_backup
```
 - If the recovery NNMi management server does not qualify for a same-system restore but meets the requirements listed in ["Different System Restore" on the previous page](#), run a command similar to the following example:


```
nmrestore.ovpl -force
```

```
-source nnmi_backups\periodic\newest_backup
```

Update the licensing as needed.

Back up Data Before Changing the Configuration

Perform scoped backups (as described in ["Backup Scope" on page 223](#)) as needed before beginning configuration changes. In this way, if your configuration changes do not have the expected effect, you will be able to revert to a known working configuration. For 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
```

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. For 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 ["Different System Restore" on page 227](#) are met. Then run a command similar to the following example:

```
nnmrestore.ovpl -lic -source nnmi_backups\offline\newest_backup
```

Restore File System Files Only

To overwrite NNMI files without affecting the database tables, run a command similar to the following example:

```
nnmrestore.ovpl -partial
-source nnmi_backups\offline\newest_backup
```

The command is useful when the NNMI implementation uses Oracle for the main NNMI database.

Backing up and Restoring the Embedded Database Only

NNMI provides the `nnmbakupembdb.ovpl` and `nnmrestoreembdb.ovpl` commands to back up and restore the NNMI embedded database only. This functionality is useful for creating a snapshot of the data as you experiment with NNMI configuration settings. The `nnmbakupembdb.ovpl` and `nnmrestoreembdb.ovpl` commands perform online backups only. At a minimum, the `nmsdbmgr` service must be running.

See the `nnmbakup.ovpl` reference page, or the Linux manpage, for more information.

Each backup operation stores files in a parent directory called `nnm-bak-<TIMESTAMP>` inside the target directory. You can specify a `-noTimestamp` option to save disk space. If you use the `-noTimestamp` option, the parent directory is simply named `nnm-bak`. When a backup is performed after a previous backup using the

-noTimestamp option, the previous backup is renamed `nnm-bak.previous`, thereby creating a rolling backup. This renaming is done after the second backup is completed to protect against any loss of backup data.

Note: Run the `nnmresetembdb.ovpl` command before restoring data to the embedded database. This command ensures that the database does not contain any errors, thereby eliminating the possibility of encountering database constraint violations. For information about running the embedded database reset command, see the `nnmresetembdb.ovpl` reference page, or the Linux manpage.

Using Backup and Restore Tools in a High Availability (HA) Environment

This section includes helpful tips to consider when using backup and restore tools in a High Availability environment.

Best Practices for Backup in an HA Environment

When using the NNMi backup tool in an HA environment, note the following best practices:

- Perform a backup using the active (primary) system. (A backup of the backup (secondary) node is not recommended because configuration files could be out-of-date and no shared disk information would be included because the backup node cannot access to the shared disk.)
- The shared disk must be connected to the active node. If using a cron job, verify that the shared disk is mounted.
- Put the system into maintenance mode (so as not to trigger a failover).
- Perform an online backup using the `nnmbackup.ovpl` script on the active node only.
- Periodically, run an offline backup.

See the `nnmbackup.ovpl` reference page, or the Linux manpage, for more information.

Best Practices for Restore in an HA Environment

When using the NNMi restore tool in an HA environment, note the following best practices

- Verify that the shared disk is mounted.
- Verify that the system is in maintenance mode.
- Perform the restore using the `nnmrestore.ovpl` script.

See the `nnmrestore.ovpl` reference page, or the Linux manpage, for more information.

For more information on using NNMi in an HA environment, see ["Configuring NNMi in a High Availability Cluster" on page 150](#).

NNMi Logging

This section describes the NNMi log file format and how to change log file properties to log sign-in and sign-out activity:

- ["NNMi Log Files" on the next page](#)
- ["Sign-in and Sign-out Logging" on the next page](#)

Also see ["NNMi Auditing" on page 112](#) for information about changing the audit log files.

NNMi Log Files

To investigate HPE Network Node Manager i Software (NNMi) performance, or to observe how NNMi processes and services are behaving, you can view log files that show a history of process and service activity. These files are available at the following location:

- *Windows*: %NnmDataDir%\log\nnm\
- *Linux*: \$NnmDataDir/log/nnm

NNMi stores these log files in a *name*.log file name format. Any archived log file has a number appended to it in the form *name*.log.%g.

- *name* is the log file base name.
- %g relates to the archive number of the archived log file. The highest appended archive number represents the oldest file.

A log file can become an archived log file after the size of the log file exceeds the configured limit. After a log file exceeds the configured limit, the last active log file is archived. For example, after NNMi archives the nnm.log file as the nnm.log.1 file, NNMi begins logging to a new nnm.log file.

NNMi logs messages at the following logging levels:

- SEVERE: Events that relate to abnormal NNMi behavior.
- WARNING: Events that indicate potential problems and all messages included in the SEVERE logging level.
- INFO: Messages written to the NNMi console (or its equivalent) and all messages included in the WARNING logging level.

Sign-in and Sign-out Logging

NNMi 10.30 is not configured to generate a log entry for each user that signs in to or out of the NNMi console. If you want to configure NNMi to log sign-in and sign-out activity, do the following:

1. Edit the following file:
 - *Windows*: %NnmDataDir%\shared\nnm\conf\props\nnm-logging.properties
 - *Linux*: \$NnmDataDir/shared/nnm/conf/props/nnm-logging.properties
2. Search for the text block containing the following line:


```
com.hp.ov.nnm.log.signin.level = OFF
```
3. Modify the line to read as follows:


```
com.hp.ov.nnm.log.signin.level = INFO
```
4. Save your changes.
5. Restart the NNMi management server:
 - a. Run **ovstop** on the NNMi management server.
 - b. Run **ovstart** on the NNMi management server.

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 `ovsstop` and `ovsstart` commands. See ["Maintenance Mode" on page 177](#) for more information.

Changing the Management Server

You can duplicate the HPE Network Node Manager i Software configuration on another system, for example, to move from a test environment to a production environment or to change the hardware of the NNMi management server.

You can change the IP address of the NNMi management server without affecting the NNMi configuration.

This chapter contains the following topics:

- ["Best Practices for Preparing the NNMi Configuration to be Moved" below](#)
- ["Moving the NNMi Configuration and Embedded Database" on the next page](#)
- ["Moving the NNMi Configuration" on the next page](#)
- ["Restoring the NNMi Public Key Certificate" on page 234](#)
- ["Changing the IP Address of a Standalone NNMi Management Server" on page 236](#)
- ["Changing the Hostname or Domain Name of an NNMi Management Server" on page 237](#)
- ["Changing the Oracle Database Instance Connection Information" on page 237](#)
- ["Changing the Password that NNMi uses to Connect to the Oracle Database Instance " on page 239](#)

Best Practices for Preparing the NNMi Configuration to be Moved

The following best practices apply to moving the NNMi configuration to a different system:

- If the node group configuration uses hostnames to identify managed nodes, the production and test NNMi management servers must use the same DNS servers. In the case that the production and test systems use different DNS servers, changes in the resolved name for a managed node might result in different polling settings between the two NNMi management servers.
- You can limit the configuration export to a single author. Create a new author value that is unique to your group or company. Specify this author value when you create or modify any of the following items:
 - Device profile
 - Incident configuration
 - URL action
- If you plan to install Smart Plug-ins (iSPIs), see the appropriate NNM iSPI document. Documentation for all NNM iSPIs is available on the HPE Software Product Manuals web site at <http://support.openview.hp.com/selfsolve/manuals>.

Moving the NNMi Configuration and Embedded Database

To move the NNMi configuration and the embedded database, for example from a test system to a production system, perform 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 NNMidatabase after the backup is made, stop all NNMi processes and create an offline backup. For example:

```
nnmbackup.ovpl -type offline -scope all -target nnm_backups\offline
```

Ensure that the requirements listed in ["Different System Restore" on page 227](#) are met on the new system, and then run a command similar to the following example:

```
nnmrestore.ovpl -source nnm_backups\offline\newest_backup
```

Caution: NNMi uses the same SSL certificate for accessing the database (embedded or external) and 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 ["Moving the NNMi Configuration" below](#), and then begin data collection fresh on the target system.

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.

Caution: Do not edit a file exported with the `nnmconfigexport.ovpl` script before using the `nnmconfigimport.ovpl` script to import the file.

For information about these commands, see the appropriate reference pages, or the Linux manpages.

Tip: The `nnmconfigexport.ovpl` command does not retain SNMPv3 credentials. For more information, see the `nnmconfigexport.ovpl` reference page, or the Linux manpage.

Note: You can only move the NNMi configuration. HPE does not support moving topology or incident data from one NNMi management server to a different NNMi management server. Nor does HPE support moving iSPI data, such as performance data that was collected for the NNM iSPI Performance for Metrics.

Restoring the NNMi Public Key Certificate

Caution: If the NNMi management server participates in NNMi application failover or is a member of a high availability (HA) cluster, contact your support representative for assistance.

The `nmn.keystore` file stores the public key certificate that NNMi uses for encryption. The NNMi installation process creates the `nmn.keystore` file and links the certificate in this file to the `nms_sec_key` record in the NNMi database (Postgres or Oracle).

If NNMi is subsequently uninstalled, but the Oracle user and database tables for NNMi are not deleted (cascaded delete of the Oracle user) before a subsequent reinstall, the `nms_sec_key` entry is not valid for the newly created `nmn.keystore` file.

To restore the NNMi public key certificate, complete the following tasks:

"Task 1: Determine the Status of KeyManager Service" below

"Task 2: Back up the Current `nmn.keystore` File" below

"Task 3: Attempt to Locate the Original `nmn.keystore` File" on the next page

"Task 4: If Available, Restore the Original `nmn.keystore` File" on page 236

Task 1: Determine the Status of KeyManager Service

1. Run the following command:

```
ovstatus -v ovjboss
```

2. In the command output, verify that the KeyManager service is not running, which usually indicates that the `nmn.keystore` file is corrupt or missing.

Note: If the `ovstatus` output shows that the KeyManager service is started, contact your support representative for assistance.

Task 2: Back up the Current `nmn.keystore` File

1. Change to the directory that contains the NNMi truststore:

Windows: %NnmDataDir%\shared\nnm\certificates

Linux: \$NnmDataDir/shared/nnm/certificates

2. For backup purposes, save copies of the following files:

`nmn.keystore`

`nmn.truststore`

Task 3: Attempt to Locate the Original nnm.keystore File

1. Determine the fingerprint of the security key in the NNMi database:

- For the embedded PostgreSQL database, enter the following:

- *Windows:*

```
%NnmInstallDir%\nonOV\Postgres\bin\psql -U postgres
-d nnm -c "<database_command>"
```

- *Linux:*

```
$NnmInstallDir/nonOV/Postgres/bin/psql -U postgres
-d nnm -c "<database_command>"
```

Replace <database_command> with the following SQL command string:

```
select fingerprint from nms_sec_key;
```

- For an Oracle database, ask the Oracle database administrator to run the <database_command> (described for the embedded database earlier in this step) in the appropriate Oracle administration tool.

The command results should be a single database row. The correct nnm.keystore file also contains this fingerprint.

2. Identify a backup nnm.keystore file to test.

This file might be in a backup of the NNMi management server in the original installation directory.

3. Test the fingerprint of a backup nnm.keystore file:

- a. Change to the directory that contains the NNMi certificates:

Windows: %NnmDataDir%\shared\nnm\certificates

Linux: \$NnmDataDir/shared/nnm/certificates

- b. Examine the contents of the keystore:

- *Windows:*

```
%jdkdir%\bin\keytool -list
-keystore nnm.keystore
```

- *Linux:*

```
$jdkdir/bin/keytool -list
-keystore nnm.keystore
```

When prompted for the keystore password, enter: nnmkeypass

The keystore output is of the form:

```
Keystore type: jks
```

```
Keystore provider: SUN
```

```
Your keystore contains 1 entry
```

```
selfsigned, Oct 28, 2008, keyEntry,
```

```
Certificate fingerprint (MD5): 29:02:D7:D7:D7:D7:29:02:29:02:29:02:29:02:29:02
```

- c. Compare the value of the MD5 fingerprint from this `nmn.keystore` file with the fingerprint in the NNMI database (from step 1 of this task).
 - o If the fingerprints match exactly, you have located a good `nmn.keystore` file for this NNMI database. Continue with ["Task 4: If Available, Restore the Original nmn.keystore File"](#) below.
 - o If the fingerprints do not match exactly, perform this task with a different `nmn.keystore` file.

Note: If you cannot locate the original `nmn.keystore` file using the above procedure, contact your support representative for assistance. Do not continue with ["Task 4: If Available, Restore the Original nmn.keystore File"](#) below.

Task 4: If Available, Restore the Original nmn.keystore File

If you located the correct `nmn.keystore` file, restore that file by following these steps:

1. Stop the NNMI management server.
Run the `ovstop` command on the NNMI management server.
2. Copy the located `nmn.keystore` file on top of the existing file in the following location:
Windows: `%NnmDataDir%\shared\nnm\certificates`
Linux: `$NnmDataDir/shared/nnm/certificates`
3. Start the NNMI management server:
Run the `ovstart` command on the NNMI management server.
4. Run the following command:
`ovstatus -v ovjboss`
5. In the command output, verify that the KeyManager service is started.

After you have verified that NNMI is working correctly, you can remove the backup copy of the `nmn.keystore` file from ["Task 2: Back up the Current nmn.keystore File"](#) on page 234.

Changing the IP Address of a Standalone NNMI Management Server

To change the IP address of the NNMI management server, follow these steps:

1. Navigate to `http://www.webware.hp.com`.
2. Log in; then follow the prompts to obtain the license key for the new IP address.
3. Copy the new license key into a text file named `license.txt`.
4. At the command prompt, enter the following command:
`nmmlicense.ovpl NNM -f license.text -nosync`
`ovstop`
5. Configure the NNMI management server with the new IP address.
6. Configure the DNS servers to recognize the new IP address of the NNMI management server.
7. Reboot the NNMI management server.
8. At a command prompt, enter the following command:

```
nmmlicense.ovpl NNM -g
```

9. In the **Autopass: License Management** dialog box, click **Remove License Key**.
10. Select the license key attached to the old IP address to remove.
11. Select **Remove Licenses permanently**.
12. Click **Remove**; then close the dialog box.

Changing the Hostname or Domain Name of an NNMi Management Server

Note: If the NNMi management server participates in NNMi application failover or is a member of a high availability (HA) cluster, contact your support representative for assistance.

To change the hostname, the domain name, or both, of the NNMi management server, set NNMi to use the new Fully Qualified Domain Name (FQDN) of the NNMi management server using the `nmsetofficialfqdn.ovpl` command. For example:

```
nmsetofficialfqdn.ovpl newnnmi.servers.example.com
```

For more information, see the `nmsetofficialfqdn.ovpl` reference page, or the Linux manpage.

Note: The FQDN is a hostname combined with a domain name. If you change either of these, you are changing the FQDN of the NNMi management server. 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 have a new SSL certificate with matching CN. The `nmsetofficialfqdn.ovpl` command updates the FQDN of the NNMi management server and it also creates a new self-signed certificate, which matches the new FQDN. However, if you are using CA certificates, you must generate a new CA certificate. See ["Generating a CA-Signed Certificate" on page 249](#) for more information.

If you change the IP address of the NNMi management server (regardless of whether the FQDN changes), you must obtain a new license. See ["Changing the IP Address of a Standalone NNMi Management Server" on the previous page](#) for more information.

Changing the Oracle Database Instance Connection Information

NNMi can be connected to one Oracle database instance at a time. You can configure this connection.

Reasons to change the Oracle database instance connection information include the following:

- The Oracle database server name must be changed.
- The port for connecting to the database conflicts with another process, or corporate policies require the use of a non-default port.
- The database instance must be renamed (for example, to meet corporate policies).
- The Oracle database server hardware must be changed.

To change the Oracle database instance that NNMi uses, complete the following tasks:

["Task 1: Update the Oracle Database Instance" on the next page](#)

"Task 2: Update the NNMi Configuration" below

Task 1: Update the Oracle Database Instance

1. Stop the NNMi management server:
Run the `ovstop` command on the NNMi management server
2. Prepare the Oracle database by moving the database, renaming the Oracle database server, or other necessary changes.
3. Verify that the target Oracle database instance meets the following prerequisites:
 - The database instance exists.
 - The database instance is populated with current NNMi data.
 - Use Oracle tools to copy NNMi data from the working database instance to the target database instance.
 - The database instance is running.

Task 2: Update the NNMi Configuration

1. Back up the database connection configuration file:

Change to the following directory:

Windows: %NnmInstallDir%\nonOV\jboss\nms\server\nms\

Linux: \$NnmInstallDir/nonOV/jboss/nms/server/nms/

Within the `nms` directory, create a directory called `deploy.save`.

Copy the `nms-ds.xml` file from the `deploy` directory to the `deploy.save` directory.

Caution: At startup, the `ovjboss` process reads all files in the `deploy` directory hierarchy. For this reason, save backup copies of the deployed files in a location outside of the `deploy` directory hierarchy, as shown in this example using the `deploy.save` directory.

2. Edit the database connection configuration file:

Change to the `deploy` directory.

In any text editor, open the `nms-ds.xml` file.

Locate the `connection-url` entry.

For example:

```
<connection-url>jdbc:oracle:thin:@ohost:1521:nnmidb1</connection-url>
```

The last three parameters in this entry are of interest. They are of the format `oracle_hostname:database_port:database_instance_name`

Change one or more of the fourth, fifth, and sixth parameters in the `connection-url` entry.

For example:

To point to a different Oracle database server, change `ohost` to another hostname.

To connect to the Oracle database server on a different port, change `1521` to another port number.

To connect to a different Oracle database instance, change `nnmidb1` to another database instance name.

Note: This database instance must already exist.

Save the `nms-ds.xml` file.

3. Start the NNMi management server:

Run the `ovstart` command on the NNMi management server.

Changing the Password that NNMi uses to Connect to the Oracle Database Instance

If you change the Oracle configuration to use a different password for connecting to the NNMi database instance, update the NNMi configuration by following these steps:

1. Stop the NNMi management server:

Run the `ovstop` command on the NNMi management server.

2. Run the `nnmchangedbpw.ovp1` command and follow the prompts.

3. Start the NNMi management server:

Run the `ovstart` command on the NNMi management server.

For more information, see the `nnmchangedbpw.ovp1` reference page, or the Linux manpage.

Modifying the Embedded Database Port

If you want to configure NNMi to use a different port for the embedded database, 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. Look for a line that resembles the following:

```
#!com.hp.ov.nms.postgres.port=5432
```

3. Uncomment the property:

```
com.hp.ov.nms.postgres.port=5432
```

Tip: To uncomment a property, remove the `#!` characters from the beginning of a line.

4. Change the existing value to the new port number.
5. Save your changes.
6. Restart the NNMi management server.
 - a. Run the `ovstop` command on the NNMi management server.
 - b. Run the `ovstart` command on the NNMi management server

Schedule Outages

NNMi lets you schedule outages for an arbitrary set of nodes using the `nmmscheduledoutage.ovpl` command. 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 one node.

See the `nmmscheduledoutage.ovpl` reference page, or the Linux manpage for more information.

Tip: See the NNMi help for more information about using the NNMi console to schedule outages.

NNMi Self Monitoring

NNMi performs self-monitoring checks, including memory, CPU, and disk resources. NNMi generates an incident after the NNMi management server becomes low on resources or detects a serious condition.

To view NNMi health information, use one of the following methods:

- From the NNMi console, select **Help > System Information**; then click the **Health** tab.
- For a detailed self-monitoring report, select **Help > NNMi System Information > Health** and click **View Detailed Health Report (Support)**.
- Run the `nmhealth.ovpl` script.

NNMi displays a status message at the bottom of the NNMi console and on the top of forms after NNMi detects a self-monitoring health exception.

To disable this warning message, complete the following steps:

1. Open the following file:
 - *Windows:* `%NNM_PROPS\nms-ui.properties`
 - *Linux:* `$NNM_PROPS/nms-ui.properties`
2. Locate the text block containing the following line:


```
#!com.hp.nms.ui.health.disablewarning=false
```
3. Uncomment and edit the following line to read as follows:


```
com.hp.nms.ui.health.disablewarning==true
```
4. Save your changes.
5. Restart the NNMi management server.
 - a. Run the `ovstop` command on the NNMi management server.
 - b. Run the `ovstart` command on the NNMi management server

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. See "[Maintenance Mode](#)" on page 177 for more information.

Chapter 7: Advanced Configuration

This section contains the following topics:

Apply Licenses

If you do not have a permanent license key installed, the NNMi product includes a temporary Instant-On license key that is valid for 60 days after you install NNMi. This temporary Instant-On license key enables you to use NNMi Ultimate features. You should obtain and install a permanent license key as soon as possible.

Note: If you have purchased NNMi Premium or NNMi Ultimate, you need to use the license keys you requested from the HPE Password Delivery Center for use with application failover or high availability. Be sure to request the following:

- **High Availability:** Obtain a license key for the virtual IP address of the NNMi HA resource group. This license key is initially used on the primary server and then used on the secondary server when needed.
- **Application Failover:** Obtain two license keys; one for the physical IP address of the primary server and one for the physical IP address of the standby server.

To view a list of the features included with an NNMi Ultimate license, see the licensing section of the *HPE NNMi Software Release Notes*.

Preparing to Install a Permanent License Key

The temporary Instant-On license has a 250 node limit. If you have been running NNMi with the Instant-On license key, you might be managing more nodes than your permanent license supports.

When tracking license information, note the following:

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

For details about license limits, see “Track Your NNMi Licenses” in the NNMi Help for Administrators.

- 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

¹VMware ESX and VMware ESXi software uses SOAP protocol to implement bare-metal hypervisors.

nodes.

For more information, see the NNMi online help.

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 HPE Network Node Manager i Software**.
2. In the **About HPE Network Node Manager i Software** window, click **Licensing Information**.
3. Look for the value shown in the **Consumption** field. This is the number of nodes that NNMi is currently managing.

When tracking license information, note the following:

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

For details about license limits, see “Track Your NNMi Licenses” in the NNMi Help for Administrators.

For details about license limits, see “Track Your NNMi Licenses” in the NNMi Help for Administrators.

4. If your permanent license supports fewer nodes than NNMi is currently managing, use the NNMi console to delete less important nodes. For more information, see *Delete a Node* in the NNMi help.

Obtaining and Installing a Permanent License Key

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

- The Entitlement Certificate, which contains the HPE product number and order number
- The IP address of one of the NNMi management servers
- If the license is for NNMi running under HA, the virtual IP address of the NNMi HA resource group
- Your company or organization information

Using Autopass and your HPE Order Number (not possible behind a firewall)

To obtain and install a permanent license key, follow these steps:

1. At a command prompt, enter the following command to open the Autopass user interface:


```
nnmlicense.ovpl NNM -gui
```
2. On the left side of the Autopass window, click **License Management**.
3. Click **Install License Key**.
4. Click **Retrieve/Install License Key**.
5. Enter your HPE Order Number and follow the Autopass prompts to complete the license key retrieval

¹VMware ESX and VMware ESXi software uses SOAP protocol to implement bare-metal hypervisors.

process.

6. NNMi automatically completes the installation.

From the Command Line

If the automated process does not run to completion (for example, if the NNMi management server is behind a firewall), follow these steps:

1. To obtain a license key, go to the HPE password delivery service at <https://webware.hpe.com/welcome.asp>
2. At a command prompt on the NNMi management server, enter the following command to update the system and to store license data files:

```
nnmlicense.ovpl NNM -flicense_file
```

(The product license ID (NNM) is case-sensitive.)

See the *nnmlicense.ovpl* reference page, or the Linux manpage, for more information.

3. NNMi automatically completes the installation.

Obtaining Additional License Keys

Contact your HPE Sales Representative or your Authorized Hewlett Packard Enterprise Reseller for information about the NNMi licensing structure, and to learn how to add license tiers for enterprise installations.

To obtain additional license keys, go to the HPE License Key Delivery Service:

<https://webware.hpe.com/welcome.asp>

See *Extend a Licensed Capacity* in the NNMi help for more information.

Note to Developers: With the NNMi Developer Toolkit, you can enhance the capabilities of NNMi by integrating custom web-service clients. After you install an NNMi Developer license, NNMi creates the `sdk-dev-kit.jar` file located in the `doc` folder. Unpack the `sdk-dev-kit.jar` file to view the NNMi Developer Toolkit documentation and samples.

Managing Certificates

Note: NNMi 10.20 introduces 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 10.30 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 "[Configure an Upgraded NNMi Environment to Use the New Keystore](#)" on page 245.

If you have upgraded to NNMi 10.30 and did not complete the steps in "[Configure an Upgraded NNMi Environment to Use the New Keystore](#)" on page 245, skip to "[Using Certificates with the JKS Repository](#)" on page 265.

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 `nm-trust.p12` file contains certificates from other parties that you expect to communicate with, or from Certificate Authorities that you trust to identify other parties. NNMi includes a self-signed certificate in both of the `nm-key.p12` and `nm-trust.p12` files.

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 `nmcertmerge.ovp1` script to merge these certificates into the `nm-key.p12` and `nm-trust.p12` files. This chapter also contains instructions to replace an expired certificate with a new self-signed or CA-signed certificate.

An administrator can disable HTTP and other unencrypted access from the network to NNMi. See ["Configure NNMi to Require Encryption for Remote Access" on page 204](#).

This chapter contains the following topics:

- ["About NNMi Certificates" below](#)
- ["Replacing an Existing Certificate with a new Self-Signed or CA-Signed Certificate" on page 256](#)
- ["Working with Certificates in Application Failover Environments" on page 257](#)
- ["Working with Certificates in High-Availability Environments" on page 258](#)
- ["Working with Certificates in Global Network Management Environments" on page 259](#)
- ["Configuring an SSL Connection to the Directory Service" on page 262](#)

About NNMi Certificates

This section describes useful terminology to help you work with certificates. Familiarize yourself with the terms mentioned in the following table.

Certificate Terminology

Concept	Description
Keystore and Truststore	<p>Truststore: NNMi truststore is the file in which you store public keys from sources that you want NNMi to trust.</p> <p>In a newly installed instance of NNMi, the name of the truststore file is <code>nm-trust.p12</code>.</p> <p>Note: On a management server where NNMi was upgraded to the version 10.30 from an older version, the truststore file name is <code>nm.truststore</code>. You can, however, perform additional steps (described in "Configure an Upgraded NNMi Environment to Use the New Keystore" on the next page) to migrate the <code>nm.truststore</code> file to the <code>nm-trust.p12</code> file.</p> <p>Keystore: NNMi keystore is the file in which you import NNMi server's private key.</p> <p>In a newly installed instance of NNMi, the name of the keystore file is <code>nm-key.p12</code>.</p> <p>Note: On a management server where NNMi was upgraded to the version 10.30 from an older version, the keystore file name is <code>nm.keystore</code>. You can, however, perform additional steps (described in "Configure an Upgraded NNMi Environment to Use the New Keystore" on the next page) to migrate the <code>nm.keystore</code> file to the <code>nm-key.p12</code> file.</p>

Certificate Terminology, continued

Concept	Description
	<p>These files are located at:</p> <ul style="list-style-type: none"> Linux: \$NNM_DATA/shared/nnm/certificates/ Windows: %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 <code>nnmkeytool.ovpl</code> utility (which uses Java's Keytool utility). Additionally, NNMi provides the <code>nnmmergecert.ovpl</code> utility to merge certificates to establish trust within NNMi systems. This program is used in HA, Failover, and GNM-RNM setups.
Supported encryption algorithms	NNMi accepts certificates generated using RSA algorithm. DSA algorithm is not supported.
Self-Signed Certificate	<p>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.</p> <p>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.</p>
CA-Signed Certificate	<p>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).</p> <p>Note: These certificates might be in a single file or in a two separate files.</p>
Root CA Certificate	Identifies the certificate authority that is trusted to sign certificates for servers and users.
Intermediate CA Certificate	<p>A certificate signed by either a root or intermediate CA that is itself an authority, rather than a server or user.</p> <p>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.</p>

Configure an Upgraded NNMi Environment to Use the New Keystore

Prior to the version 10.20, NNMi used to provide a Java KeyStore (JKS) repository to store certificates. NNMi 10.20 introduces 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 10.20 on a system.

However, when you upgrade an older version of NNMi (older than 10.20) to the version 10.30, 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:

- *On Windows:*

```
%nnminstalldir%\bin\nnmkeytool.ovpl -importkeystore -srckeystore
%nnmdatadir%\shared\nnm\certificates\nnm.keystore -destkeystore
%nnmdatadir%\shared\nnm\certificates\nnm-key.p12 -srcstoretype JKS -deststoretype
PKCS12 -srcprovidername SUN -destprovidername PKCS12 -alias <src_alias>
```

- *On 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 PKCS12 -
srcprovidername SUN -destprovidername PKCS12 -alias <src_alias>
```

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.

3. Run the following command to migrate to the new truststore file:

- *On Windows:*

```
%nnminstalldir%\bin\nnmkeytool.ovpl -importkeystore -srckeystore
%nnmdatadir%\shared\nnm\certificates\nnm.truststore -destkeystore
%nnmdatadir%\shared\nnm\certificates\nnm-trust.p12 -srcstoretype JKS -deststoretype
PKCS12 -srcprovidername SUN -destprovidername PKCS12
```

- *On 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 PKCS12
```

4. Open the server.properties file from the following location with a text editor:

- *On Windows:*

```
%nnmdatadir%\nmsas\nms
```

- *On 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=${com.hp.ov.DataDir}
/shared/nnm/certificates/nnm-key.p12
nmsas.server.keystore.cred=nnmkeypass
nmsas.server.security.truststore.file=${com.hp.ov.DataDir}
/shared/nnm/certificates/nnm-trust.p12
nmsas.server.truststore.cred=ovpass
nmsas.server.security.keystore.alias==
nms.comm.soap.https.PROTOCOLS=TLSv1.2
```

Note: If you want to discover and monitor virtual networks running on VMware ESXi 5.1 servers, set the `nms.comm.soap.https.PROTOCOLS` property to `TLSv1`, `TLSv1.1`, `TLSv1.2`.

7. Save the file.
8. Open the `nms-local.properties` file from the following location with a text editor:

- *On Windows:*

```
%nmdataDir%\conf\nnm\props
```

- *On Linux:*

```
/var/opt/OV/conf/nnm/props
```

9. Modify the values of all the javax parameters:

Parameter	Value
<code>javax.net.ssl.trustStore</code>	<code>\${NnmDataDir}/shared/nnm/certificates/nnm-trust.p12</code>
<code>javax.net.ssl.trustStoreType</code>	PKCS12
<code>javax.net.ssl.keyStore</code>	<code>\${NnmDataDir}/shared/nnm/certificates/nnm-key.p12</code>
<code>javax.net.ssl.keyStoreType</code>	PKCS12

10. Save the file.
11. Delete the `nnm.keystore` and `nnm.truststore` files from the following directory
- *On Windows:*

```
%nmdataDir%\shared\nnm\certificates
```
 - *On Linux:*

```
/var/opt/OV/shared/nnm/certificates
```
12. Restart NNMI.

Using Certificates with the PKCS #12 Repository

Prior to the version 10.20, NNMi used to provide a Java KeyStore (JKS) repository to store certificates. NNMi 10.20 introduces 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 10.20 on a system.

However, when you upgrade an older version of NNMi (older than 10.20) to the version 10.30, 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.

To check the type of certificate repository:

1. Log on to the NNMi console.
2. Click **Help > System Information**, and then go to the Server tab.
3. Check the value of the `javax.net.ssl.keyStore` property.
 - If the property points to the `nnm-key.p12` file, your environment has a PKCS#12 repository.
 - If the property points to the `nnm.keystore` file, your environment has a JKS repository.

Alternatively, do the following:

1. On the NNMi management server, as root or administrator, run the following command:
 - *On Windows:* `%nnminstalldir%\bin\nnmprops -l`
 - *On Linux:* `/opt/OV/bin/nnmprops -l`
2. From the command output, note the value of the `javax.net.ssl.trustStoreType` property.
 - The value of this property indicates the type of certificate repository.

Generating a Self-Signed Certificate

Note: NNMi 10.20 introduces 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 10.20 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 ["Configure an Upgraded NNMi Environment to Use the New Keystore" on page 245](#).

If you have upgraded to NNMi10.30 and did not complete the steps in ["Configure an Upgraded NNMi Environment to Use the New Keystore" on page 245](#), skip to ["Generating a Self-Signed Certificate" on page 267](#).

To generate a self-signed certificate, follow these steps:

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. 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 3650 -keyalg rsa -keystore nnm-key.p12 -storetype PKCS12 -storepass nnmkeypass -alias <alias_name>
 - Linux: \$NnmInstallDir/bin/nnmkeytool.ovpl -genkeypair -validity 3650 -keyalg rsa -keystore nnm-key.p12 -storetype PKCS12 -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, HPE 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.

Caution: 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 ["Generating a CA-Signed Certificate" below](#).

HPE recommends that you use CA-signed certificates.

Generating a CA-Signed Certificate

Note: NNMi 10.20 introduces 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 10.30 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 ["Configure an Upgraded NNMi Environment to Use the New Keystore" on page 245](#).

If you have upgraded to NNMi 10.30 and did not complete the steps in ["Configure an Upgraded NNMi Environment to Use the New Keystore" on page 245](#), skip to ["Generating a CA-Signed Certificate" on page 268](#).

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

1. Generate a self-signed certificate. For details, see ["Generating a Self-Signed Certificate" on the previous page](#).
2. Run the following command to create a CSR (Certificate Signing Request) file:

- **Windows:** `%NnmInstallDir%\bin\nnmkeytool.ovpl -keystore nnm-key.p12 -certreq -storetype PKCS12 -storepass nnmkeypass -alias <alias_name> -file CERTREQFILE`
- **Linux:** `$NnmInstallDir/bin/nnmkeytool.ovpl -keystore nnm-key.p12 -certreq -storetype PKCS12 -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.

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 ["Types of CA-Signed Certificates" on page 253](#).

The CA signing authority returns one of the following:

- A single signed server certificate file (referred to as `myserver.crt` 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` 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.

Note: 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.

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. Additionally, if you installed iSPIs on the NNMI management server, you must import the server-signed certificate too to the truststore file.

Note: iSPIs that reside on the NNMI management server use NNMI's certificates.

- If you received a single file from [step 3](#)
 - i. Copy the root CA certificates from that file into a separate `myca.crt` file.
 - ii. *(Only if you installed iSPIs on the NNMI management server)* Copy the server certificate (the NNMI certificate that is CA-signed) from that file into a separate `nnmi-server.crt` file.
 - If you received a set of two files from [step 3](#)
 - i. *(Only if you installed iSPIs on the NNMI management server)* Save a copy of the `myserver.crt` file as `nnmi-server.crt`.
 - ii. 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.p12 -storetype PKCS12 -storepass nnmkeypass -file <path_to_myserver.crt>
```

Linux:

```
$NnmInstallDir/bin/nnmkeytool.ovpl -importcert -trustcacerts -keystore
nnm-key.p12 -storetype PKCS12 -storepass nnmkeypass -file <path_to_myserver.crt>
```

Note: 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.

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:D7:29:02:29:02:29:02:29:02:29:02

SHA1: C4:03:7E:C4:03:7E:C4:03:7E:C4:03:7E:C4:03:7E:C4:03

Trust this certificate? [no]: y

Certificate was added to keystore

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 PKCS12
```

- Linux:

```
$NnmInstallDir/bin/nnmkeytool.ovpl -list -keystore nnm-trust.p12 -storetype PKCS12
```

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

Example truststore output

The truststore output is of the form:

```
Keystore type: pkcs
```

```
Keystore provider: JKS
```

```
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
```

Tip: The truststore can include multiple certificates.

11. Import the server certificate (the NNMi certificate that is CA-signed) into the truststore file.

Note: Follow this step only if you installed iSPi on the NNMi management server.

Run the following commands to import the CA-signed server certificate into the truststore file:

Windows:

- `%NnmInstallDir%\bin\nnmkeytool.ovpl -import -alias <alias_name> -storetype PKCS12 -keystore nnm-trust.p12 -file <path_to_nnmi-server.crt> -storepass ovpass`

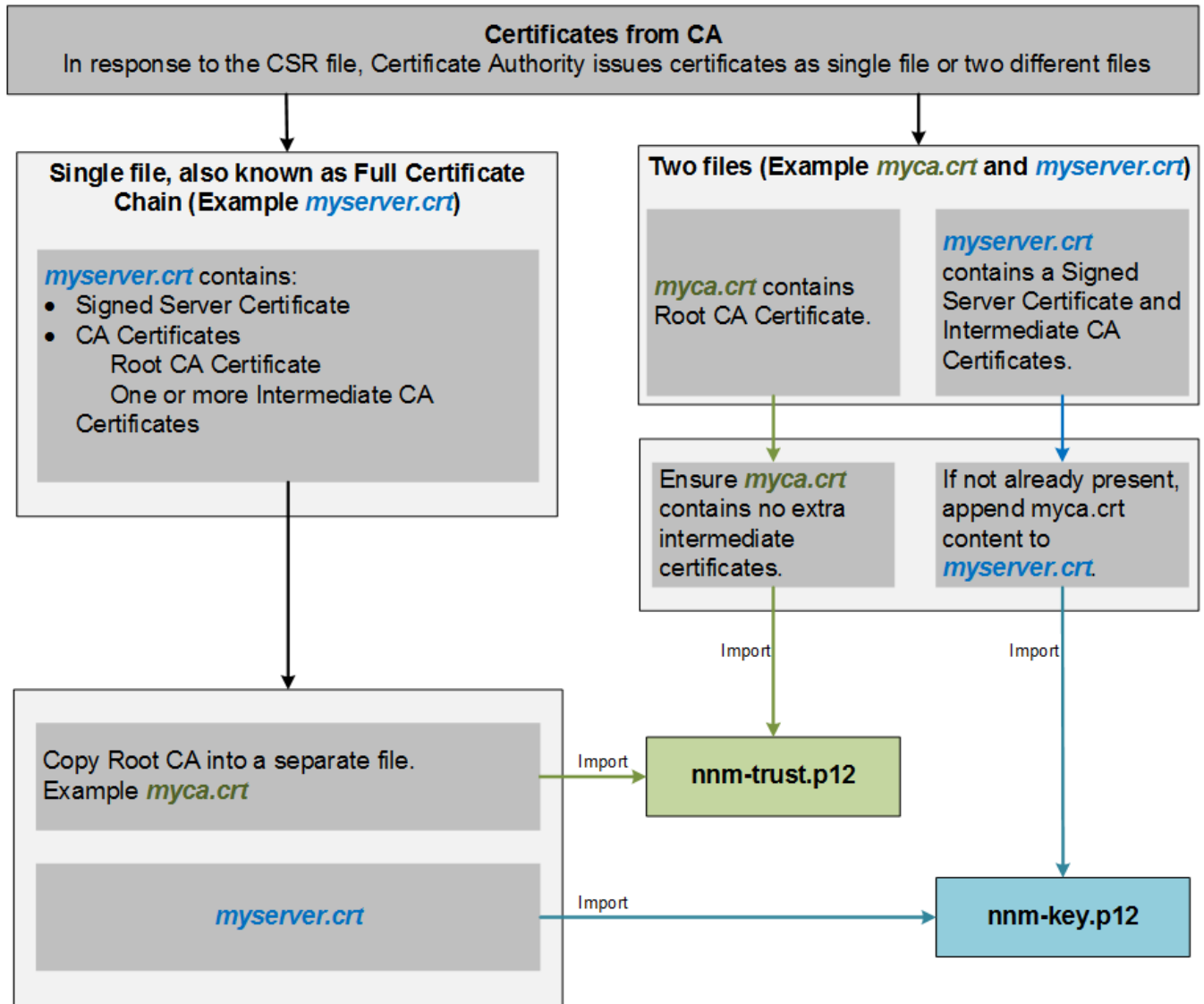
Linux:

- `$NnmInstallDir/bin/nnmkeytool.ovpl -import -alias <alias_name> -storetype PKCS12 -keystore nnm-trust.p12 -file <path_to_nnmi-server.crt> -storepass ovpass`

Note:

- In the above command,
- `<path_to_nnmi-server.crt>` corresponds to the full path of the location where you have stored the server certificate (the NNMi certificate that is CA-signed).
 - `<alias_name>` corresponds to the alias you had provided at the time of generating the certificate.

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`.

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.

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**.

- 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 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` and the root CA Certificate into the `nnm-trust.p12`. 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.p12` 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. Also, be sure to remove any extra intermediate certificates from the `myca.crt` file. This should result in the following files:

- `myserver.crt`, containing the full certificate chain
- `myca.crt`, containing the root CA Certificate

Note: When using a CA, only the root CA certificate is generally added to the `nnm-trust.p12`. Adding intermediate CA or server certificates to the `nnm-trust.p12` 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` 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/AVQQKExNQ0EgQ29ycG9yYXRpb24gTHRkMRAwDgYDVQQLew0ZXR3b3J3s
eGV5ZXZvY2F0aW9uTG1zdD9iYXNlP29iamVjdENsYXNzPWNSTERpc3RyaWJ1dG1w
.....
.....
TZImiZPyLGQBGRYDaW50MRIwEAYKZCZImiZPyLGQBGRYCC2cxZzARBgNVBAMTCmNb
pSo6o/76yShtT7Vr1fz+mXjWyeHAiy/QLCpPebYhejHEg4dZgzWWT/1Qt==
```

```
-----END CERTIFICATE-----
```

Combined server and CA certificates in one file:

```
-----BEGIN CERTIFICATE-----
```

```
Sample1/VQQKExNQ0EgQ29ycG9yYXRpb24gTHRkMRAwDgYDVQLEwdOZXR3b3Js
eGV5ZSZZvY2F0aW9uTG1zdD9iYXNlP29iamVjdENSYXNzPWNSTERpc3RyaWJ1dG1w
```

```
.....
.....
```

```
TZImiZPyLGQBGRYDaW50MRIwEAYKZCZImiZPyLGQBGRYCC2cxZzARBgNVBAMTCmNb
pSo6o/76yShtT7Vr1fz+mXjWyEHaIy/QLCpPebYhejHEg4dZgzWWT/1Qt==
```

```
-----END CERTIFICATE-----
```

```
-----BEGIN CERTIFICATE-----
```

```
Sample2/Gh0dHA6Ly9jb3JwMWRjc2cyLnNnLm1udC5wc2FnbG9iYWwuY29tL0Nlc
Ra0CApwwggKYMB0GA1UdDgQWBBSqaWZzCRcpvJW0FPZ/Be9b+QSPyDAfBgNVHSMC
```

```
.....
.....
```

```
Wp5Lz1ZJA0u1VHbPVdQnXn1Bkx7V65niLoaT90Eqd61a1iV1JHj7GBrij90uvVGu
BQagggEChoG9bGRhcDovLy9DTj1jb3JwMWRjc2cyL==
```

```
-----END CERTIFICATE-----
```

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 `nmm-key.p12` and `nmm-trust.p12` files:
 - *Windows:* `%NnmDataDir%\shared\nnm\certificates`
 - *Linux:* `$NnmDataDir/shared/nnm/certificates`
2. Save a backup copy of the `nmm-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 nmm-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.p12 -storetype PKCS12 -storepass nnmkeypass -alias <alias>
 - *Linux:* \$NnmInstallDir/bin/nnmkeytool.ovpl -delete -keystore nnm-key.p12 -storetype PKCS12 -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

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 ["Generating a Self-Signed Certificate" on page 248](#). Or, if your 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 ["Generating a CA-Signed Certificate" on page 249](#).
2. Delete the existing certificate from NNMi keystore by following the instructions in ["Delete a Certificate from the NNMi Keystore" on the previous page](#).
3. 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.

Working with Certificates in Application Failover Environments

Using Certificates with Application Failover



Note: NNMi10.30 introduces 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 NNMi10.30 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 ["Configure an Upgraded NNMi Environment to Use the New Keystore" on page 245](#).

If you have upgraded to NNMi10.30 and did not complete the steps in ["Configure an Upgraded NNMi Environment to Use the New Keystore" on page 245](#), skip to ["Working with Certificates in Application Failover Environments" on page 273](#).

When configuring the application failover feature, you must merge the content of the truststore file for both nodes into one `nmm-trust.p12` 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 `nmm-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 `nmm-trust.p12` file.

Windows:

```
nmmcertmerge.ovpl -truststore <truststore>
```

Linux:

```
nnmcertmerge.ovpl -truststore <truststore>
```

- 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

- 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, rather redo 257 through 258.

Windows:

```
%NnmInstallDir%\bin\nnmkeytool.ovpl -list -keystore
%NnmDataDir%\shared\nnm\certificates\nnm-trust.p12
-storetype PKCS12 -storepass ovpass
```

Linux:

```
$NnmInstallDir/bin/nnmkeytool.ovpl -list -keystore
$NnmDataDir/shared/nnm/certificates/nnm-trust.p12 -storetype PKCS12 -storepass ovpass
```

- Continue configuring the application failover feature at "[Configuring NNMi for Application Failover](#)" on page 122.

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.

Using Certificates with HA



Note: NNMi10.30 introduces 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 NNMi10.30 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 "[Configure an Upgraded NNMi Environment to Use the New Keystore](#)" on page 245.

If you have upgraded to NNMi10.30 and did not complete the steps in ["Configure an Upgraded NNMi Environment to Use the New Keystore"](#) on page 245, skip to ["Working with Certificates in High-Availability Environments"](#) on page 275.

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.

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.

Note: 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 ["Maintenance Mode"](#) on page 177 for more information.

Tip: You can complete this procedure before or after configuring NNMi for HA, as described in ["Shared NNMi Data in High Availability Environments"](#) on page 172.

- Change to the following directory on NNMi_HA1 before completing step 2:
 - Windows:* `%NnmDataDir%\shared\nnm\certificates`
 - Linux:* `$NnmDataDir/shared/nnm/certificates`
- On NNMi_HA1, run the following commands to import `newcert` into the `nnm-key.p12` file:
 - Windows:* `%NnmInstallDir%\bin\nnmkeytool.ovpl -import -alias newcert_Alias -storetype PKCS12 -keystore nnm-key.p12 -file newcert`
 - Linux:* `$NnmInstallDir/bin/nnmkeytool.ovpl -import -alias newcert_Alias -storetype PKCS12 -keystore nnm-key.p12 -file newcert`

Working with Certificates in Global Network Management Environments

Note: NNMi10.30 introduces 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 NNMi10.30 on a system. Environments upgraded from an older version of NNMi continue to use a JKS repository to store certificates.

If you have upgraded to NNMi 10.30 and did not complete the steps in ["Configure an Upgraded NNMi Environment to Use the New Keystore"](#) on page 245, skip to ["Configuring Certificates in Global Network Management Environments"](#) on page 276.

In upgraded environments, you can migrate to the PKCS #12 repository by using the steps in ["Configure an Upgraded NNMi Environment to Use the New Keystore"](#) on page 245.

Configuring Certificates in Global Network Management Environments

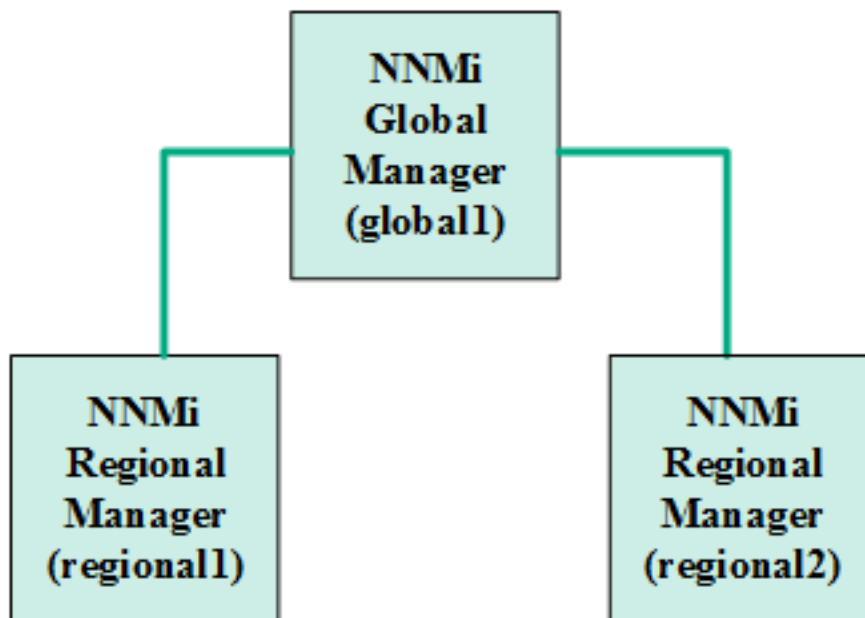
During NNMi installation, the installation script creates a self-signed certificate for the NNMi management server. This certificate contains an alias that includes the fully-qualified domain name of the node. The installation script adds this self-signed certificate to the NNMi management server's `nmm-key.p12` and `nmm-trust.p12` 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 ["Replacing an Existing Certificate with a new Self-Signed or CA-Signed Certificate"](#) on page 256.

Note: If you are using a mix of newly installed NNMi 10.30 instances and NNMi management servers upgraded to the version 10.30 from an older version, follow the guideline in ["Configure an Upgraded NNMi Environment to Use the New Keystore"](#) on page 245.

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 command on `global1` to merge the `regional1` and `regional2` certificates into `global1`'s `nnm-trust.p12` file.

Windows:

- a. `nnmcertmerge.ovpl -truststore regional1_nnm-trust.p12_location`
- b. `nnmcertmerge.ovpl -truststore regional2_nnm-trust.p12_location`

Linux

- a. `nnmcertmerge.ovpl -truststore regional1_nnm-trust.p12_location`
 - b. `nnmcertmerge.ovpl -truststore regional2_nnm-trust.p12_location`
4. Run the following command sequence on `global1`:
 - a. Run `ovstop` on the `global1` NNMi management server.
 - b. Run `ovstart` on the `global1` NNMi management server.

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. See "[Maintenance Mode](#)" on page 177 for more information.

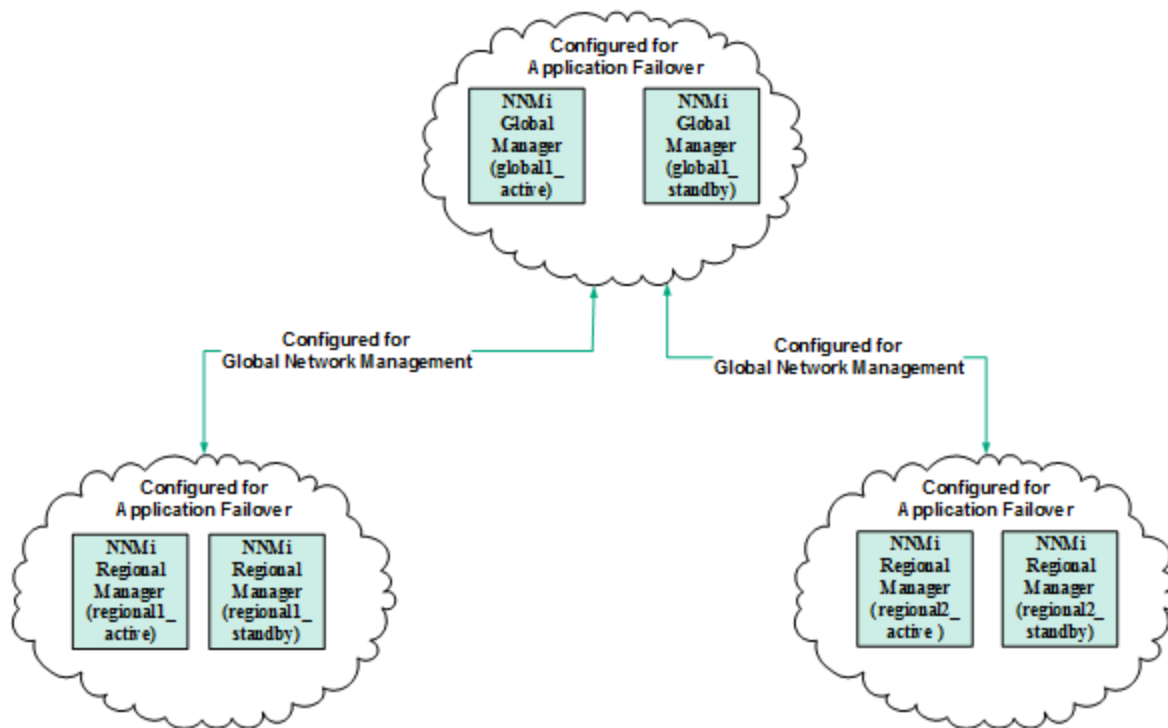
Configuring Certificates in Global Network Management Environments with Failover

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.p12` and `nnm-trust.p12` files.

Note: If you are using a mix of newly installed NNMi 10.30 instances and NNMi management servers upgraded to the version 10.30 from an older version, follow the guideline in "[Configure an Upgraded NNMi Environment to Use the New Keystore](#)" on page 245.

This example uses the global network management configuration with the application failover feature as shown in the following diagram:

Global Network Management with Application Failover



Complete the following steps to configure the global network management feature to work with application failover based on the above diagram.

1. Follow the instructions shown in ["Working with Certificates in Application Failover Environments"](#) on [page 257](#) for each application failover cluster shown in the above diagram.
2. Complete the configuration for application failover shown in ["Application Failover Requirements"](#) on [page 123](#).
3. Follow the instructions shown in ["Configuring Certificates in Global Network Management Environments"](#) on [page 260](#) for `regional1_active` and `regional2_active`.

Configuring an SSL Connection to the Directory Service

Note: NNMi10.30 introduces 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 NNMi10.30 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 ["Configure an Upgraded NNMi Environment to Use the New Keystore" on page 245](#).

If you have upgraded to NNMi10.30 and did not complete the steps in ["Configure an Upgraded NNMi Environment to Use the New Keystore" on page 245](#), skip to ["Configuring an SSL Connection to the Directory Service" on page 278](#).

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.
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.

- a. Run the following command:

- o *Windows:*

```
%NnmInstallDir%\bin\nnmkeytool.ovpl -import
-alias nnmi_ldap -storetype PKCS12 -keystore nnm-trust.p12
-file <Directory_Server_Certificate.txt>
```

- o *Linux:*

```
$NnmInstallDir/bin/nnmkeytool.ovpl -import
-alias nnmi_ldap -storetype PKCS12 -keystore nnm-trust.p12
-file <Directory_Server_Certificate.txt>
```

Where <Directory_Server_Certificate.txt> is your company's truststore certificate.

- b. When prompted for password, enter: **ovpass**
- c. 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:C4:03:7E: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
```

- *Linux:*

```
$NnmInstallDir/bin/nnmkeytool.ovpl -list
-storetype PKCS12 -keystore nnm-trust.p12
```

When prompted for the keystore 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

```

Tip: The truststore can include multiple certificates.

5. Restart the NNMi management server.

- Run the `ovstop` command on the NNMi management server.
- Run the `ovstart` command on the NNMi management server.

Note: 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 "[Maintenance Mode](#)" on page 177 for more information.

Using Certificates with the JKS Repository

Prior to the version 10.20, NNMi used to provide a Java KeyStore (JKS) repository to store certificates. NNMi 10.20 introduces 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 10.20 on a system.

However, when you upgrade an older version of NNMi (older than 10.20) to the version 10.30, 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 if your NNMi environment uses the PKCS#12 repository.

To check the type of certificate repository:

1. Log on to the NNMi console.
2. Click **Help > System Information**, and then go to the Server tab.
3. Check the value of the `javax.net.ssl.keyStore` property.

If the property points to the `nnm-key.p12` file, your environment has a PKCS#12 repository.

If the property points to the `nnm.keystore` file, your environment has a JKS repository.

Alternatively, do the following:

1. On the NNMi management server, as root or administrator, run the following command:
 - *On Windows:* `%nnminstalldir%\bin\nnmprops -l`
 - *On Linux:* `/opt/OV/bin/nnmprops -l`
2. From the command output, note the value of the `javax.net.ssl.trustStoreType` property.
The value of this property indicates the type of certificate repository.

Many tasks in this section require you to use the `keytool` utility. The location of the `keytool` utility on the NNMi management server depends on the type of JDK configured with NNMi. To find out the location of the `keytool` utility:

1. On the NNMi management server, as root or administrator, run the following command:
 - *On Windows:* `%nnminstalldir%\bin\nnmprops -l`
 - *On Linux:* `/opt/OV/bin/nnmprops -l`
2. From the command output, note the value of the property `com.hp.ov.nms.jdk.dir`. This value indicates the home directory of JDK.

The `keytool` utility resides within the `bin` directory under this JDK home directory.

Tip: To be able to access the `keytool` utility easily, you can create an environment variable (for example, `jdkdir`) that points to this JDK home directory.

About NNMi JKS Certificates

Certificate Terminology

Concept	Description
Keystore and Truststore	<p>Truststore: NNMi truststore is the <code>nnm.truststore</code> file in which you store public keys from sources that you want NNMi to trust.</p> <p>Keystore: NNMi keystore is the <code>nnm.keystore</code> file in which you import NNMi server's private key.</p> <p>The <code>nnm.truststore</code> and <code>nnm.keystore</code> files are located at:</p> <ul style="list-style-type: none"> Linux: <code>\$NNM_DATA/shared/nnm/certificates/</code> Windows: <code>%NNM_DATA%\shared\nnm\certificates\</code>
Default NNMi certificates	<p>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.</p>
Tools	<p>Certificates are generated and managed using Java's Keytool utility. Additionally, NNMi provides the <code>nnmmergecert.ovp1</code> utility to merge certificates to establish trust within NNMi systems. This program is used in HA, Failover, and GNM-RNM setups.</p>
Supported encryption algorithms	<p>NNMi accepts certificates generated using RSA algorithm. DSA algorithm is not supported.</p>
Self-Signed Certificate	<p>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.</p> <p>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.</p>
CA-Signed Certificate	<p>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).</p> <p>Note: These certificates might be in a single file or in a two separate files.</p>
Root CA Certificate	<p>Identifies the certificate authority that is trusted to sign certificates for servers and users.</p>
Intermediate CA Certificate	<p>A certificate signed by either a root or intermediate CA that is itself an authority, rather than a server or user.</p> <p>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.</p>

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 ["Generating a Self-Signed Certificate" below](#).
2. If your 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 ["Generating a CA-Signed Certificate" on the next page](#) ["Generating a CA-Signed Certificate" on page 249](#)
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%\nsm\props\nms-local.properties`
 - *Linux:* `$NNM_CONF/nsm/props/nms-local.properties`
4. Restart the NNMi Management Server.
 - a. Run the `ovstop` command on the NNMi management server.
 - b. Run the `ovstart` command on the NNMi management server.

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. See ["Maintenance Mode" on page 177](#) for more information.

5. Test HTTPS access to the NNMi console using the following syntax:

https://<fully_qualified_domain_name>:<port_number>/nsm/.

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.

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 `nsm.keystore` and `nsm.truststore` files:

- Windows: %NnmDataDir%\shared\nnm\certificates
 - Linux: \$NnmDataDir/shared/nnm/certificates
2. Save a backup copy of the nnm.keystore file.

Note:

- 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 3650 -keyalg rsa - keystore nnm.keystore -storepass nnmkeypass - alias <alias_name>
 - Linux: \$jdkdir/bin/keytool -genkeypair -validity 3650 -keyalg rsa -keystore nnm.keystore -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, HPE 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.

Caution: 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 ["Generating a CA-Signed Certificate" below](#).

HPE recommends that you use CA-signed certificates.

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 ["Generating a Self-Signed Certificate" on the previous page](#).
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:

- In the command above, <alias_name> corresponds to the alias you had provided at the time of generating the certificate.
- For more information about the `keytool` command, search for “Key and Certificate Management Tool” at <http://www.oracle.com/technetwork/java/index.html>.

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 "[Types of CA-Signed Certificates](#)" on page 271.
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 -storepass 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:D7:29:02:29:02:29:02:29:02:29:02
SHA1: C4:03:7E:C4:03:7E:C4:03:7E:C4:03:7E:C4:03:7E:C4:03
Trust this certificate? [no]: y
Certificate was added to keystore
```

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 -
file <myca.crt>
```

- *Linux:*

```
$jdkdir/bin/keytool -import -alias <alias_name> -keystore nnm.truststore -file <myca.crt>
```

Note:

- In the above command,
 - `<myca.crt>` corresponds to the full path of the location where you have stored the CA certificates.
 - `<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 -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

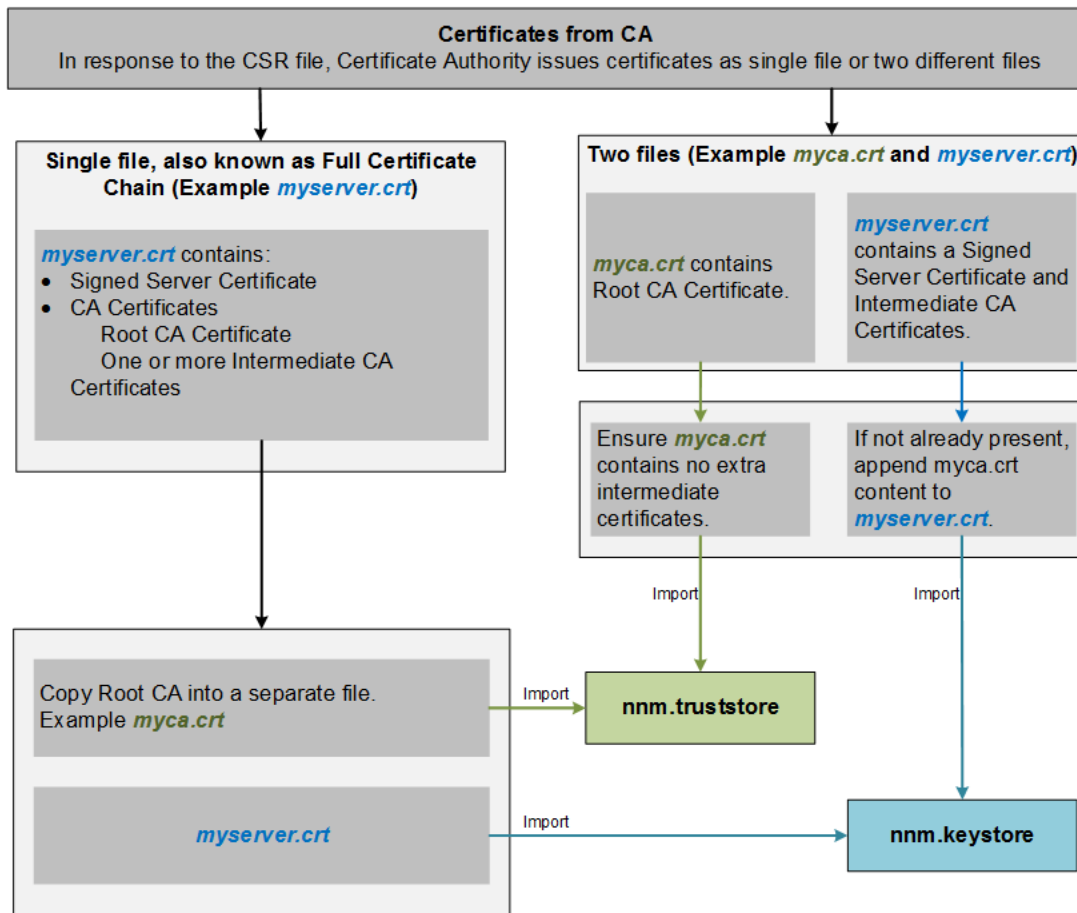
Your keystore contains 1 entry

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

Tip: The truststore can include multiple certificates.

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*.
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.

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**.

- 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 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.

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.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/AVQQKExNQU0EgQ29ycG9yYXRpb24gTHRkMRAwDgYDVQQLEwd0ZXR3b3Js
eGV5ZXZvY2F0aW9uTG1zdD9iYXNlP29iamVjdENSYXNzPWNSTERpc3RyaWJ1dG1w
.....
.....
TZImiZPyLQGBGRYDaW50MRIwEAYKCZImiZPyLQGBGRYCC2cxEzARBgNVBAMTCmNb
pSo6o/76yShtT7Vr1fz+mXjWyEHaIy/QLCpPebYhejHEg4dZgzWWT/1Qt==
-----END CERTIFICATE-----
```


Combined server and CA certificates in one file:

```
-----BEGIN CERTIFICATE-----
Sample1/VQQKExNQ0EgQ29ycG9yYXRpb24gTHRkMRAwDgYDVQQLewd0ZXR3b3Js
eGV5ZlZvY2F0aw9uTG1zdD9iYXNlP29iamVjdENSyXNzPWNSTERpc3RyaWJ1dG1w
.....
.....
TZImiZPyLGQBGRYDaW50MRIwEAYKcZImiZPyLGQBGRYCC2cxZARBgNVBAMTCmNb
pSo6o/76yShtT7Vr1fz+mXjWyEHaIy/QLCPebYhejHEg4dZgzWWT/1Qt==
-----END CERTIFICATE-----

-----BEGIN CERTIFICATE-----
Sample2/Gh0dHA6Ly9jb3JwMWRjc2cyLnNnLm1udC5wc2FnbG9iYWwuY29tL0N1c
Ra0CApwwggKYYMB0GA1UdDgQWBBSqawZzCRcpvJW0FPZ/Be9b+QSPyDAfBgNVHSMC
.....
.....
Wp5Lz1ZJA0u1VHbPVdQnXn1Bkx7V65niLoat90Eqd61aliV1JHj7GBriJ90uvVGu
BQagggEChoG9bGRhcDovLy9DTj1jb3JwMWRjc2cyL==
-----END CERTIFICATE-----
```

Working with Certificates in Application Failover Environments

Using Certificates with Application Failover



When configuring the application failover feature, you must merge the content of the `nnm.keystore` and `nnm.truststore` files for both nodes into one `nnm.keystore` file and one `nnm.truststore` 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).

- Change to the following directory on Server Y :
 - Windows:* %NnmDataDir%\shared\nnm\certificates
 - Linux:* \$NnmDataDir/shared/nnm/certificates
- 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>.
- Run the following command on Server X to merge Server Y's certificates into Server X's nnm.keystore and nnm.truststore files.

Windows:

```
nmcertmerge.ovpl -keystore <keystore> -truststore <truststore>
```

Linux:

```
nmcertmerge.ovpl -keystore <keystore> -truststore <truststore>
```
- 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:* %NnmDataDir%\shared\nnm\certificates
 - Linux:* \$NnmDataDir/shared/nnm/certificates
- 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, rather redo [274](#) through [274](#).

Windows:

```
%jdkdir%\bin\keytool.exe -list -keystore  
%NnmDataDir%\shared\nnm\certificates\nnm.keystore -storepass nmkeypass
```

Linux:

```
$jdkdir/bin/keytool -list -keystore $NnmDataDir/shared/nnm/certificates/nnm.keystore  
-storepass nmkeypass
```
- 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, rather redo [274](#) through [274](#).

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
```
- Continue configuring the application failover feature at "[Configuring NNMi for Application Failover](#)" on [page 122](#).

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.

Using Certificates with HA



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.

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.

Note: 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 ["Maintenance Mode" on page 177](#) for more information.

Tip: You can complete this procedure before or after configuring NNMi for HA, as described in ["Shared NNMi Data in High Availability Environments" on page 172](#).

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.keystore` file:
 - *Windows:* %jdkdir%\bin\keytool -import -alias `newcert_Alias` -keystore `nnm.keystore` -file `newcert`

- *Linux*: `$jdkdir/bin/keytool -import -alias newcert_Alias -keystore nnm.keystore -file newcert`
3. Edit the following file on both the active (NNMi_HA1) and the standby (NNMi_HA2) nodes:
 - *Windows*: `%NnmDataDir%\conf\nnm\props\nms-local.properties`
 - *Linux*: `$NnmDataDir/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.

Working with Certificates in Global Network Management Environments

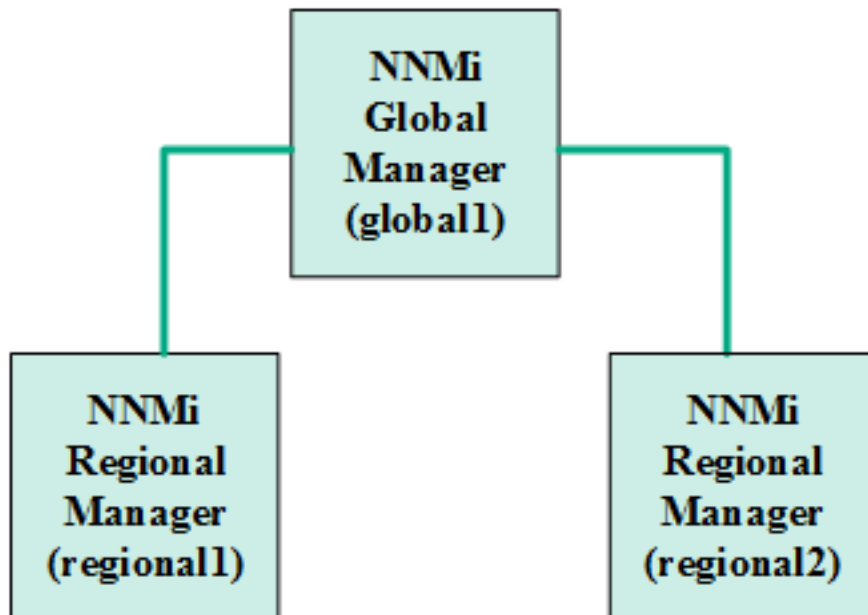
Configuring Certificates in Global Network Management Environments

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.

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 ["Replacing an Existing Certificate with a New Self-Signed or CA-Signed Certificate" on page 267](#).

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.truststore` files from the above locations on `regional1` and `regional2` to some temporary location on `global1`.
3. Run the following command on `global1` to merge the `regional1` and `regional2` certificates into `global1`'s `nnm.truststore` file.

Windows:

 - a. `nnmcertmerge.ovpl -truststore regional1_nnm.truststore_location`
 - b. `nnmcertmerge.ovpl -truststore regional2_nnm.truststore_location`

Linux

 - a. `nnmcertmerge.ovpl -truststore regional1_nnm.truststore_location`
 - b. `nnmcertmerge.ovpl -truststore regional2_nnm.truststore_location`
4. Run the following command sequence on `global1`:
 - a. Run `ovstop` on the `global1` NNMi management server.
 - b. Run `ovstart` on the `global1` NNMi management server.

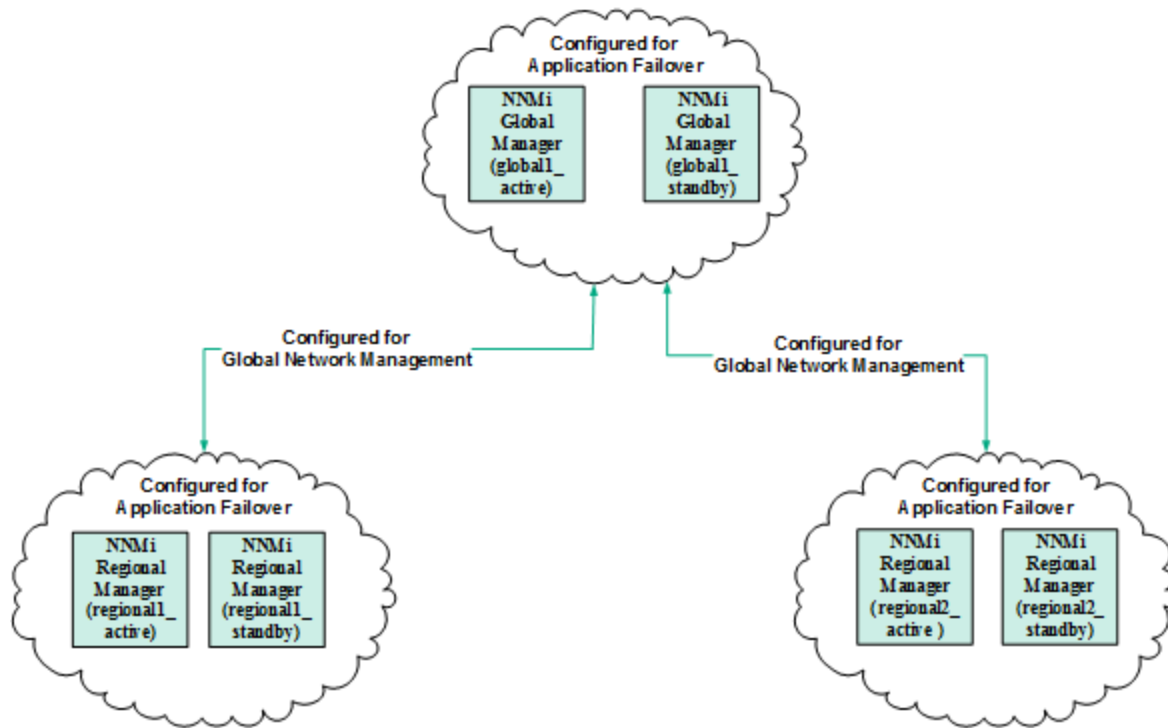
Note: When making file changes under High Availability (HA), you need to make the changes on both nodes in the cluster. For NNMi using HA configurations, if the change requires you to stop and restart the NNMi management server, you must put the nodes in maintenance mode before running the `ovstop` and `ovstart` commands. See "[Maintenance Mode](#)" on page 177 for more information.

Configuring Certificates in Global Network Management Environments with Failover

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.

This example uses the global network management configuration with the application failover feature as shown in the following diagram:

Global Network Management with Application Failover



Complete the following steps to configure the global network management feature to work with application failover based on the above diagram.

1. Follow the instructions shown in ["Working with Certificates in Application Failover Environments" on page 273](#) for each application failover cluster shown in the above diagram.
2. Complete the configuration for application failover shown in ["Application Failover Requirements" on page 123](#).
3. Follow the instructions shown in ["Working with Certificates in Global Network Management Environments" on page 276](#) for regional1_active and regional2_active.

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.

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.
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 the root CA certificate of the LDAP directory server (without intermediate certificates) into the NNMi truststore:

- a. Run the following command:

- *Windows:*

```
%jdkdir%\bin\keytool.exe -import
-alias nmi_ldap -keystore nnm.truststore
-file <Directory_Server_Certificate.txt>
```

- *Linux:*

```
$jdkdir/bin/keytool -import
-alias nmi_ldap -keystore nnm.truststore
-file <Directory_Server_Certificate.txt>
```

Where <Directory_Server_Certificate.txt> is your company's truststore certificate.

- b. When prompted for the keystore password, enter: **ovpass**
- c. 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:C4:03:7E:C4:03
Trust this certificate? [no]: y
Certificate was added to keystore
```

4. 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 keystore 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
```

Tip: The truststore can include multiple certificates.

5. Restart the NNMi management server.
 - a. Run the `ovstop` command on the NNMi management server.
 - b. Run the `ovstart` command on the NNMi management server.

Note: 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 "[Maintenance Mode](#)" on page 177 for more information.

For more information about the `keytool` command, search for "Key and Certificate Management Tool" at <http://www.oracle.com/technetwork/java/index.html>.

Using Single Sign-On (SSO) with NNMi

You can configure HPE Network Node Manager i Software (NNMi) single sign-on (SSO) to facilitate access to NNM iSPIs from the NNMi console. With SSO, when you log on to the NNMi console, you receive access to NNM iSPIs and other HPE applications without needing to log on again. SSO provides easier access to NNM iSPIs and other HPE applications while maintaining a secure level of access. After you sign out of the NNMi console (or the NNMi console session times out), you must re-enter your sign-in credentials to access NNM iSPI and other HPE application URLs outside the NNMi console.

SSO is not enabled during installation. If it was, browsing from one NNMi management server to another logs you out of the first one, providing little benefit. To keep this from happening, SSO is initially disabled so you can coordinate setting the `initString` and `protectedDomains` parameter among the NNMi management servers, as explained in this chapter.

This chapter contains the following topics:

- "[SSO Access for NNMi](#)" on the next page
- "[Enabling SSO for a Single Domain](#)" on page 282
- "[Enabling SSO for NNMi Management Servers Located in Different Domains](#)" on page 282
- "[SSO Access for NNMi and the NNM iSPIs](#)" on page 283

- ["Disabling SSO" on page 284](#)
- ["SSO Security Notes" on page 285](#)

SSO Access for NNMi

To browse among several NNMi management servers, you must do one of the following:

Note: 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 ["Maintenance Mode" on page 177](#) for more information.

- Edit the `nms-ui.properties` file and make the parameter values for `com.hp.nms.ui.sso.initString` and `com.hp.nms.ui.sso.protectedDomains` the same among the NNMi management servers. Make sure to set the `com.hp.nms.ui.sso.domain` parameter to match the domain an NNMi management server resides in.
 - If you have NNMi management servers residing in only one network domain, follow the instructions show in ["Enabling SSO for a Single Domain" on the next page](#).
 - If you have NNMi management servers residing in more than one network domain, follow the instructions shown in ["Enabling SSO for NNMi Management Servers Located in Different Domains" on the next page](#) for more information.
- Edit the `nms-ui.properties` file and make sure you have SSO disabled. See ["Disabling SSO" on page 284](#) for more information.

If you choose to not complete one of these actions, each time you browse to a different NNMi management server, you will be automatically signed out of the previous NNMi management server.

There are special considerations for using SSO with the NNMi global network management feature. See ["Configuring Single Sign-On for Global Network Management" on page 381](#) for more information.

If the domain name of the NNMi management server is short, as in `mycompany`, without any period (`.`), the NNMi console will immediately sign you out. The restrictions for SSO browser cookies require a domain name to contain at least one period, such as `mycompany.com`. To remedy this situation, complete the following steps:

1. Open the following file in a text editor:
 - *Windows:* `%NNM_PROPS%/nms-ui.properties`
 - *Linux:* `$(NNM_PROPS)/nms-ui.properties`
2. For this example, search for the following string:


```
com.hp.nms.ui.sso.domain = mycompany
```

 and replace it with the following string:


```
com.hp.nms.ui.sso.domain = mycompany.com
```
3. Run the following command to commit the changes:

```
nnmssso.ovpl -reload
```

See the `nnmssso.ovpl` reference page, or the Linux manpage, for more information.

Enabling SSO for a Single Domain

To enable SSO for use in a single domain, complete the following steps:

Note: 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 ["Maintenance Mode" on page 177](#) for more information.

1. Open the following file:

- *Windows:* %NNM_PROPS%\nms-ui.properties
- *Linux:* \$NNM_PROPS/nms-ui.properties

2. Look for a section in the file that resembles the following:

```
com.hp.nms.ui.sso.isEnabled = false
```

Change this as follows:

```
com.hp.nms.ui.sso.isEnabled = true
```

3. Look for a section in the file that resembles the following:

```
com.hp.nms.ui.sso.domain = mycompany.com
```

Change *mycompany.com* to the domain the NNMi management server resides in. Make sure there is only one domain listed when enabling SSO in a single domain.

4. Look for a section in the file that resembles the following:

```
com.hp.nms.ui.sso.protectedDomains = mycompany.com
```

Change *mycompany.com* to the domain the NNMi management server resides in. Make sure there is only one protected domain listed when enabling SSO in a single protected domain.

5. Run the following command to commit the changes:

```
nnmssso.ovpl -reload
```

See the *nnmssso.ovpl* reference page, or the Linux manpage, for more information.

Enabling SSO for NNMi Management Servers Located in Different Domains

You can configure two or more NNMi management servers for SSO. This example explains how to configure SSO for three NNMi management servers located in different domains. If you must configure two or more NNMi management servers for SSO and these systems reside in different domains, complete the following steps:

Note: 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 ["Maintenance Mode" on page 177](#) for more information.

1. Open the following file:
 - *Windows*: %NNM_PROPS%\nms-ui.properties
 - *Linux*: \$NNM_PROPS/nms-ui.properties
2. Look for a section in the file that resembles the following:


```
com.hp.nms.ui.sso.isEnabled = false
```

 Change this as follows:


```
com.hp.nms.ui.sso.isEnabled = true
```
3. Look for a section in the file that resembles the following:


```
com.hp.nms.ui.sso.domain = group1.mycompany.com
```

 Make sure the domain name contains at least one dot.
4. Look for a section in the file that resembles the following:


```
com.hp.nms.ui.sso.protectedDomains=group1.mycompany.com
```

 Change this as follows:


```
com.hp.nms.ui.sso.protectedDomains=group1.mycompany.com, group2.yourcompany.com, group3.yourcompany.com
```
5. Look for a section in the file that resembles the following:


```
com.hp.nms.ui.sso.initString =Initialization String
```

 NNMi management servers must share the same initialization string to work in an SSO configuration. Change the initialization string the same value on all NNMi management servers included in the SSO configuration.
6. Run the following command to commit the changes:


```
nnmssso.ovpl -reload
```

 See the *nnmssso.ovpl* reference page, or the Linux manpage, for more information.
7. Repeat [step 1](#) through [step 6](#) two more times, configuring the remaining two NNMi management servers. For each remaining NNMi management server, substitute *group2* or *group3* for *group1* during [step 3](#).

SSO Access for NNMi and the NNM iSPIs

After SSO is enabled, SSO between NNMi and the NNM iSPIs does *not* require `initString` configuration.

To use SSO, access NNMi as follows:

- Use the correct URL in the following form:


```
<protocol>://<fully_qualified_domain_name>:<port_number>/nnml <protocol>
```

`<protocol>` represents either `http` or `https`.

`<fully_qualified_domain_name>` represents the official fully-qualified domain name (FQDN) of the NNMi management server.

`<port_number>` is the port for connecting to the NNMi console, is assigned during NNMi installation, and is specified in the following file:

- *Windows:* %NnmDataDir%\conf\nnm\props\nms-local.properties
- *Linux:* \$NnmDataDir/conf/nnm/props/nms-local.properties
- Log on to NNMi using a valid account.

For SSO to work, URL access to NNMi and the NNM iSPiS must share a common network domain name. Additionally, the URL must not include an IP address. If you do not have a FQDN for the NNMi management server, you can substitute the IP address of the NNMi management server. However, doing so disables single sign-on for NNM iSPiS, and you must log on again the next time you access any NNM iSPiS.

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 official FQDN set during installation. See the `nnmofficialfqdn.ovpl` reference page, or the Linux manpage, for more information.
- In the NNMi console, click **Help > System Information**. On the **Server** tab, look for the official FQDN statement.

If you must change the official FQDN set during installation, use the `nnmsetofficialfqdn.ovpl` command. See the `nnmsetofficialfqdn.ovpl` reference page, or the Linux manpage, for more information.

Note: After installation, the system account is still valid. Use the system account only for command-line security and for recovery purposes.

SSO to NNM iSPiS require that users access the NNMi console through a URL that contains the official FQDN. You can configure NNMi to redirect NNMi URLs to the official FQDN when the NNMi console is accessed through a non-official domain name, such as an IP address or a shortened version of the domain name. Before configuring NNMi to redirect URLs, an appropriate official FQDN must be configured. For information, see the NNMi help.

After you enable NNMi to redirect URLs, note the following:

- You can log on to the NNMi console using any hostname that is valid for the NNMi management server you want to access. For example, if you request `http://localhost/nnm`, NNMi redirects you to a URL such as `http://host.mydomain.com/nnm`.
- If you cannot access the NNMi console using `http://host.mydomain.com/nnm`, use the following to directly access the NNMi console:

`<protocol>://<fully_qualified_domain_name>:<port_number>launch?cmd=showMain.`

`<protocol>` represents either `http` or `https`.

`<fully_qualified_domain_name>` represents the official fully-qualified domain name (FQDN) of the NNMi management server.

`<port_number>` is the port for connecting to the NNMi console, is assigned during NNMi installation, and is specified in the following file:

- *Windows:* %NnmDataDir%\conf\nnm\props\nms-local.properties
- *Linux:* \$NnmDataDir/conf/nnm/props/nms-local.properties

Disabling SSO

If you have a need to disable SSO, complete the following steps:

Note: 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 ["Maintenance Mode" on page 177](#) for more information.

1. Open the following file:
 - *Windows:* %NNM_PROPS%\nms-ui.properties
 - *Linux:* \$NNM_PROPS/nms-ui.properties
2. Look for a section in the file that resembles the following:

```
com.hp.nms.ui.sso.isEnabled = true
```

Change the `isEnabled` property to `false`:

```
com.hp.nms.ui.sso.isEnabled = false
```

3. Run the following command to commit the changes:

```
nnmssso.ovpl -reload
```

See the *nnmssso.ovpl* reference page, or the Linux manpage, for more information.

SSO Security Notes

1. The `initString` parameter in SSO security is used as follows:

SSO uses *Symmetric Encryption* to validate and create an SSO token. The `initString` parameter within the configuration is used for initialization of the secret key. An application creates a token, and each application that uses the same `initString` parameter validates the token.

Note: The following information is very important:

- It is not possible to use SSO without setting the `initString` parameter.
- The `initString` parameter is confidential information and should be treated as such in terms of publishing, transporting, and persistency.
- Applications that integrate with each other can share the `initString` using SSO.
- The minimum length of the `initString` is 12 characters.

2. Disable SSO unless it is specifically required.
3. The application that uses the weakest authentication framework, and issues an SSO token that is trusted by other integrated applications, determines the level of authentication security for all the applications.
HPE recommends that only applications using strong and secure authentication frameworks issue an SSO token.
4. Symmetric encryption implication:

SSO uses symmetric cryptography for issuing and validating SSO tokens. Therefore, any application using SSO can issue a token to be trusted by all other applications sharing the same `initString`.

This potential risk is relevant when an application sharing the `initString` either resides or is accessible in an untrusted location.

5. User roles:

SSO does not share user roles between integrated applications. Therefore, the integrated application must monitor user roles. HPE recommends you share the same user registry (as LDAP/AD) among all integrated applications.

Failure to manage user roles might cause security breaches and negative application behavior. For example, the same user name might be assigned to different roles in the integrated applications.

There could be situations when a user logs on to application A, then accesses application B that uses container or application authentication. The failure to manage the user role will force the user to manually log on to application B and enter a username. If the user enters a different user name than the one used to log on to application A, the following unexpected behavior can arise: If the user subsequently accesses a third application, application C, from application A or application B, then the user will access it using the user names that were used to log on to application A or application B respectively.

6. Identity Manager is used for an authentication:

All unprotected resources in the Identity Manager must be configured as nonsecure URL settings in the SSO configuration.

7. SSO demonstration mode:

- Use the SSO demonstration mode for demonstrative purposes only.
- Only use the demonstration mode in unsecured networks.
- Do not use the demonstration mode in production. Any combination of the demonstration mode with the production mode should not be used.

Configuring NNMi to Support Public Key Infrastructure User Authentication

NNMi supports user authentication through Public Key Infrastructure (PKI) so that users must log on to NNMi using an X.509 client certificate without using a password. The information in this chapter explains how to configure NNMi (using PKI user authentication) to map certificates to NNMi user accounts.

Note: PKI user authentication includes support for smart cards, such as Common Access Card (CAC) and Personal Identity Verification (PIV) cards.

After enabling NNMi to use PKI user authentication, NNMi users do not need to use an NNMi-specific user name and password to log on to NNMi.

Using this approach, NNMi reads your PKI certificate to obtain your user name. To obtain NNMi user roles, you need to define a user's roles within NNMi or configure NNMi to use Lightweight Directory Access Protocol (LDAP).

Note: PKI user authentication uses the HTTPS protocol.

Note: PKI user authentication is a replacement for the Lightweight Single Sign-on (LW-SSO) functionality. Therefore, you cannot use them both. See ["Disabling SSO" on page 284](#) for more information.

This chapter contains the following topics:

["User Authentication Strategies" below](#)

["Configuring NNMi for PKI User Authentication \(X.509 Certificate Authentication\)" below](#)

["Certificate Validation \(CRL and OCSP\)" on page 291](#)

["Validating Certificates Using CRLs" on page 293](#)

["Validating Certificates Using Online Certificate Status Protocol \(OCSP\)" on page 296](#)

["Configuring NNMi to Restrict Certificates Used for NNMi Logon Access" on page 299](#)

["Example: Configuring NNMi to Require a Smart Card Logon" on page 300](#)

["Configuring CLI Authentication for PKI User Authentication" on page 303](#)

["Troubleshooting PKI User Authentication Issues" on page 305](#)

User Authentication Strategies

NNMi provides several options for where the NNMi user access information is defined and stored.

The following table indicates the options available for PKI user authentication.

User Authentication Strategies

Option	Which Method for User Authentication?	User Account Definitions in NNMi	User Group Definitions in NNMi	Which Method for Group Membership
Mixed	X.509 Certificate	Yes	Yes	NNMi User Account Mappings
External	X.509 Certificate	No	Yes	LDAP

In the Mixed option, NNMi defines and stores the User Group assignments. For information about setting up all user information in NNMi, see **Configuring User Accounts (User Account Form)** in the NNMi help.

In the External option, NNMi uses the Lightweight Directory Access Protocol (LDAP) User Group assignments. For more information, see ["Integrating NNMi with a Directory Service through LDAP" on page 308](#).

Configuring NNMi for PKI User Authentication (X.509 Certificate Authentication)

Before configuring NNMi for PKI user authentication, note that user account names must match the user names contained in the certificates. Set roles using one of the following methods:

- To use LDAP, see ["Integrating NNMi with a Directory Service through LDAP" on page 308.](#)
- To use the NNMi console to add a user account, select the **Directory Service Account** check box on the **User Account** form and leave the **Password** field blank. Then, use the user account name to match the previous mapping rule.

For NNMi, enable and customize PKI user authentication in the following file:

- *Windows:* %NnmDataDir%\nmsas\NNM\conf\nms-auth-config.xml
- *Linux:* \$NnmDataDir/nmsas/NNM/conf/nms-auth-config.xml

To enable NNMi to require PKI user authentication, also referred to as X.509 Certificate Authentication, follow these steps:

1. Edit the following file:

- *Windows:* %NnmDataDir%\nmsas\NNM\conf\nms-auth-config.xml
- *Linux:* \$NnmDataDir/nmsas/NNM/conf/nms-auth-config.xml

2. Search for the following text block:

```
<realm name="console">
<mode>FORM</mode>
</realm>
```

3. Edit the located lines to read:

```
<realm name="console">
<mode>X509</mode>
</realm>
```

4. Search for the following text block:

```
<principalMapping>
```

5. Configure NNMi to extract (map) the principal by editing the items in the <principalMapping> section. You must know the format of your certificate to complete this step.

Note: NNMi supports several options for extracting a principal and those options can be specified in any order and in any number.

- The attribute element extracts a field from the SubjectDN; for example, EMAILADDRESS.
 - If you are using LDAP, the extracted name must match the name the LDAP configuration expects. For more information, see [Integrating NNMi with a Directory Service through LDAP.](#)
 - If you use internal accounts, the name must match the NNMi user account name. If the account is used for PKI user authentication only, it should be created as a "Directory Service Account", without a password (using the NNMi **User Account** form. Select the **Directory Service Account** check box and leave the **Password** field blank). If the account is used for both PKI user authentication and password logon, it should be created as a standard account with a password.
- The regexp element runs the regular expression against the whole SubjectDN.

- The `subjectAlternativeName` (SAN) element can be used with type `rfc822Name` (which is an email address).
- The `subjectAlternativeName` element with type `otherName` and an additional `oid` attribute. This option is commonly used for the Microsoft Universal Principal Name (UPN) field.

In addition to the examples provided in the `nms-auth-config.xml` file's `<principalMapping>` section, see the following examples:

Example 1: Edit the following lines to read as follows for using the EMAIL field:

```
<!-- The attribute element extracts a field from the SubjectDN;
for example, EMAILADDRESS, CN, or UID. -->
<attribute>EMAILADDRESS</attribute>
```

Example 2: Edit the following lines as an example of using a more complex regular expression to extract part of the field, as in extracting just part of the EMAILADDRESS field. To extract just the name part of the EMAILADDRESS field, use the following regular expression:

```
<!-- Extract the name part of the email field which appears first
in the subjectDN. If the subject is EMAILADDRESS=first.last@example.com,
CN=First Last, OU=MyGroup, O=My Company, the mapped username would be
"first.last"--> <regex group="1">EMAILADDRESS=(^[^@]+).*</regex>
```

Example 3: Edit the following lines as an example of using a more complex regular expression to match fields in the middle of the string:

```
<!--Extract the CN field which appears anywhere in the subjectDN.
Note the optional group before the CN which matches the
previous fields. If the subject is EMAILADDRESS=first.last@example.com,
CN=First Last, OU=MyGroup, O=My Company
```

Example 4: Edit the following lines to read as follows to extract the email address from the Subject Alternative Name:

```
<!-- Extract the first match of type rfc822Name from the Subject
Alternative Name field of the certificate. -->
<subjectAlternativeName type="rfc822Name" />
```

Example 5: Edit the following lines to read as follows to extract a particular OID from the Subject Alternative Name:

```
<!-- Extract the first match of type otherName with the supplied
OID from the Subject Alternative Name field of the certificate. -->
<subjectAlternativeName type="otherName" oid="1.3.6.1.4.1.311.20.2.3" />
```

Note: The logging command to enable debug logging is as follows:

```
nnmsetlogginglevel.ovpl
com.hp.ov.nms.as.server.auth.x509.NmsCertMapper FINEST
```

6. Save your changes.
7. If you have already installed your trusted CA certificates into the truststore, run the following script for the changes to the `nms-auth-config.xml` file to take immediate effect:

```
nnmsecurity.ovpl -reloadAuthConfig
```

Otherwise, if you have not yet installed your certificates, proceed with the following steps.

8. Change to the directory on the NNMI management server that contains the `nnm-trust.p12` file:
 - Windows:* `%NnmDataDir%\shared\nnm\certificates`
 - Linux:* `$NnmDataDir/shared/nnm/certificates`
9. Import your trusted CA certificate into the `nnm-trust.p12` file. Suppose the `example_ca.cer` file contains the certificate you must use. Run the following command to import the CA certificate into the NNMI `nnm-trust.p12` file:

Windows:

```
%NnmInstallDir%\bin\nnmkeytool.ovpl -import -alias myca -storetype PKCS12 -keystore
nnm-trust.p12 -file example_ca.cer
```

Linux:

```
$NnmInstallDir/bin/nnmkeytool.ovpl -import -alias myca -storetype PKCS12 -keystore
nnm-trust.p12 -file example_ca.cer
```

10. Restart the NNMI services.
 - a. Run the `ovstop` command on the NNMI management server.
 - b. Run the `ovstart` command on the NNMI management server.

Note: When making file changes under HA, you must make the changes on both nodes in the cluster. For NNMI using HA configurations, if the change requires you to stop and restart the NNMI management server, you must put the nodes in maintenance mode before running the `ovstop` and `ovstart` commands.

NNMI is now configured to use PKI user authentication. You can no longer use passwords to log on NNMI. Check that your LDAP and NNMI user accounts are working correctly, and that the certificates and accounts are configured correctly for user access to NNMI.

Logging on to NNMI using a Client Certificate

To log on to NNMI using a client certificate, follow these steps:

1. Ensure that your client certificate is accessible in your browser.
2. Point your browser to `https://<hostname>/nnm`.
3. NNMI permits you access and assigns user roles based on your NNMI or LDAP account configuration.

Revoking Access for a User Having a Client Certificate

To remove a user from accessing NNMi, do one of the following:

- If you configured a user for access using an LDAP account, remove the user from all LDAP groups associated with NNMi.
- If you configured a user for access using NNMi user accounts, remove the user from the user group and remove their user account.

In either case, the user can no longer log on to the NNMi console.

Special Considerations When PKI User Authentication in Global Network Management Environments

If you use NNMi in a Global Network Management configuration, configure PKI user authentication for all of the NNMi management servers included in the Global Network Management Configuration.

Certificate Validation (CRL and OCSP)

NNMi supports two methods of checking for revoked certificates:

- Certificate Revocation List (CRL) - A CRL is a list of revoked certificates that is downloaded from the Certificate Authority (CA).
- Online Certificate Status Protocol (OCSP) - OCSP is a protocol for checking revocation of a single certificate interactively using an online service called an OCSP responder.

CRL and OCSP validation are two different ways to achieve the same result: denying access to any user whose certificate is revoked. In a web browser, OCSP is generally considered superior because a browser is usually dealing with many different Certificate Authorities (CAs), and having to download an entire CRL to check one web site is inefficient.

However, for a server that is often dealing with many clients, all with certificates from the same CA, CRL checking can be significantly more efficient because the CRL can be downloaded once per day instead of needing to check OCSP for every connection.

When both OCSP and CRL are enabled, NNMi, by default, queries CRL first. CRL checking is performed first because the CRL usually has a much longer lifetime and, therefore, is more resilient to network outages. OCSP performs frequent requests so, if the network or the OCSP responder is down, users will be unable to log on. NNMi attempts to obtain a valid CRL first to use in continuing operations in the case the network or OCSP responder goes down.

In addition, CRL comparison is much faster than OCSP; that is, matching a certificate against a list that exists on the disk is faster than querying a separate server over the network to validate each certificate. So if a certificate has been signed by a trusted entity, and is not expired, the CRL is queried to see if the certificate has been revoked. If it has been revoked, there is no need to check OCSP. But if the certificate is still valid after checking the CRL, OCSP will also be queried to ensure that the certificate has not been revoked recently (and an updated CRL listing the certificate is not yet available).

When both OCSP and CRL are enabled, NNMi supports the following:

- NNMi queries CRL first, followed by OCSP (this is the default behavior).
- If the CRL is not available, OCSP is used as a backup.
- If OCSP is not available, CRL is used as a backup.

General Configuration for Certificate Validation Protocols

You can configure how NNMi checks for revoked certificates. For example, you can configure the order in which protocols are used, and whether all the protocols are used.

NNMi uses the `nms-auth-config.xml` file to configure such settings.

Configuring Protocol Order

By default, NNMi performs CRL checking, and then OCSP checking.

To configure the order in which the certificate validation protocols check for revoked certificates, do the following:

1. Edit the following file:

Windows: %NnmDataDir%\nmsas\NNM\conf\nms-auth-config.xml

Linux: \$NnmDataDir/nmsas/NNM/conf/nms-auth-config.xml

2. Within the `<revocation>` section of the file (find the `<revocation>` tag), search for the line that begins with the following text:

```
<ordering>
```

3. Do one of the following:

- To specify that CRL checking is to be used first, followed by OCSP, edit the line to read as follows:

```
<ordering>CRL OCSP</ordering>
```

- To specify that OCSP checking is to be used first, followed by CRL, edit the line to read as follows:

```
<ordering>OCSP CRL</ordering>
```

4. Save the `nms-auth-config.xml` file.
5. Run the following command for the change to take effect:

```
nnmsecurity.ovpl -reloadAuthConfig
```

Configuring Protocol Requests

You can configure NNMi to do either of the following with regard to protocol requests:

- Check all certificate validation protocols for each certificate
- Check the protocol list in the preferred order and stop when a valid response is received

To configure protocol requests, do the following:

1. Edit the following file:

Windows: %NnmDataDir%\nmsas\NNM\conf\nms-auth-config.xml

Linux: \$NnmDataDir/nmsas/NNM/conf/nms-auth-config.xml

2. Within the `<revocation>` section of the file (find the `<revocation>` tag), search for the line that begins with the following text:

```
<mode>
```

3. Do one of the following:

- To have NNMi check all protocols for each certificate, edit the line to read as follows:

```
<mode>CHECK_ALL</mode>
```

- To have NNMi check the protocol list in the preferred order and stop when a valid response is received, edit the line to read as follows:

```
<mode>FIRST_SUCCESS</mode>
```

4. Save the `nms-auth-config.xml` file.
5. Run the following command for the change to take effect:

```
nmmsecurity.ovpl -reloadAuthConfig
```

Validating Certificates Using CRLs

NNMi uses CRLs to properly deny access to clients using a certificate that is no longer trusted.

Note: During authentication, when a certificate's serial number is found in a CRL, NNMi does not accept that certificate and authentication fails.

NNMi checks CRLs by default when using X.509 authentication mode; however, you can specify a CRL by editing the `nms-auth-config.xml` file, as described in the following sections.

Note: NNMi stores the CRL configuration in the following location:

- *Windows:* `%NnmDataDir%\nmsas\NNM\conf\nms-auth-config.xml`
- *Linux:* `$NnmDataDir/nmsas/NNM/conf/nms-auth-config.xml`

There is also a default version of the configuration file, which can be used for reference purposes to view new available options. The default configuration file is stored in the following location:

- *Windows:* `%NnmInstallDir%\newconfig\HPOvNnmAS\nmsas\conf\nms-auth-config.xml`
- *Linux:* `$NnmInstallDir/newconfig/HPOvNnmAS/nmsas/conf/nms-auth-config.xml`

Enabling and Disabling CRL Checking

By default, NNMi enables CRL checking.

To configure CRL checking, follow these steps:

1. Edit the following file:

```
Windows: %NnmDataDir%\nmsas\NNM\conf\nms-auth-config.xml
```

```
Linux: $NnmDataDir/nmsas/NNM/conf/nms-auth-config.xml
```

2. Within the `<cr1>` section of the file (find the `<cr1>` tag), search for the line that begins with the following text:

```
<enabled>
```

3. Do one of the following:

- To enable CRL checking, change the line to read as follows:

```
<enabled>>true</enabled>
```

- To disable CRL checking, change the line to read as follows:

```
<enabled>>false</enabled>
```

4. Save the `nms-auth-config.xml` file.

5. Run the following command for the change to take effect:

```
nmmsecurity.ovpl -reloadAuthConfig
```

Changing the CRL Enforcement Mode

By default, NNMi is set to enforce CRLs.

To change the product's enforcement of CRLs, follow these steps:

1. Edit the following file:

```
Windows: %NnmDataDir%\nmsas\NNM\conf\nms-auth-config.xml
```

```
Linux: $NnmDataDir/nmsas/NNM/conf/nms-auth-config.xml
```

2. Within the `<cr1>` section of the file (find the `<cr1>` tag), search for the line that begins with the following text:

```
<mode>
```

3. Change the line to read as one of the following:

```
<mode><value></mode>
```

where `<value>` is one of the following:

- ENFORCE: Enforce CRLs where specified in the certificates
- ATTEMPT: Check CRLs but allow access if the CRL is not available
- REQUIRE: Require and enforce CRLs in certificates

Note: In REQUIRE mode, authentication will fail if there is no CRL specified or available for a user's certificate.

4. Save the `nms-auth-config.xml` file.

5. Run the following command for the change to take effect:

```
nmmsecurity.ovpl -reloadAuthConfig
```

Changing How Often a CRL Should be Refreshed

To configure how often NNMi refreshes the CRL, follow these steps:

1. Edit the following file:

Windows: %NnmDataDir%\nmsas\NNM\conf\nms-auth-config.xml

Linux: \$NnmDataDir/nmsas/NNM/conf/nms-auth-config.xml

2. Within the <cr1> section of the file (find the <cr1> tag), search for the line that begins with the following text:

```
<refreshPeriod>
```

3. Change the line to read as follows:

```
<refreshPeriod><value></refreshPeriod>
```

where <value> is the integer number of hours or days (the smallest value is 1h).

For example, enter 24h for 24 hours; enter 2d for 2 days.

4. Save the nms-auth-config.xml file.
5. Run the following command for the change to take effect:

```
nnmsecurity.ovpl -reloadAuthConfig
```

Changing the Maximum Idle Time for a CRL

You can configure how long NNMi keeps a CRL after the CRL has been idle (has not been used or accessed).

To change the maximum idle time for a CRL, follow these steps:

1. Edit the following file:

Windows: %NnmDataDir%\nmsas\NNM\conf\nms-auth-config.xml

Linux: \$NnmDataDir/nmsas/NNM/conf/nms-auth-config.xml

2. Within the <cr1> section of the file (find the <cr1> tag), search for the line that begins with the following text:

```
<maxIdleTime>
```

3. Change the line to read as follows:

```
<maxIdleTime><value></maxIdleTime>
```

where <value> is the integer number of hours or days (the smallest value is 1h).

For example, enter 24h for 24 hours; enter 2d for 2 days.

4. Save the nms-auth-config.xml file.
5. Run the following command for the change to take effect:

```
nnmsecurity.ovpl -reloadAuthConfig
```

CRL Expiration Warnings

When CRL checking is enabled, if a CRL expires, users might be locked out of the NNMi console. To help avoid unwanted lockouts, NNMi provides health warning messages to alert administrators that a CRL has either expired or will be expiring soon.

The *expired* CRL warning (Major severity) occurs when one or more CRLs have expired.

The *expiring* CRL warning (Minor severity) occurs when one or more CRLs has less than 1/6th of its valid period remaining. For example, if a CRL is valid for 24 hours, NNMi displays a warning if the CRL expires in fewer than four hours.

Configure the refresh period such that CRLs are always kept fresh. A properly configured refresh period ensures that, if the CRL server is unavailable for a time, there is a sufficient valid period remaining for the downloaded CRLs. In this way, NNMi can continue normal operation until the CRL server is available. In this example, a refresh period of eight hours might be appropriate.

Changing the Location for a CRL

By default, NNMi downloads CRLs from the HTTP location embedded in the certificate. If this location is not accessible to the NNMi management server, the administrator can obtain the required CRLs some other way and configure NNMi to load those CRLs from the local file system.

Note: Only CRLs signed by the certificate issuer are considered when evaluating the certificate.

To configure NNMi to load CRLs from the local file system, do the following:

1. Edit the following file:

Windows: %NnmDataDir%\nmsas\NNM\conf\nms-auth-config.xml

Linux: \$NnmDataDir/nmsas/NNM/conf/nms-auth-config.xml

2. Within the <crl> section of the file (find the <crl> tag), search for the following text block:

```
<!--
```

Optional specification for the CRL location. If set NNMi will treat all certificates issued by the same CA as this CRL as having this CRL location. Multiple entries may be listed.

```
<location>file:///var/opt/OV/shared/nnm/certificates/myco.crl</location>
```

```
-->
```

3. Insert a line after the --> tag, and enter the following, based on your operating system:

Windows: <location>file:///C:/CRLS/<crlname>.crl</location>

Linux: <location>file:///var/opt/OV/shared/nnm/certificates/<crlname>.crl
</location>

4. Save the nms-auth-config.xml file.
5. Run the following command for the change to take effect:

```
nnmsecurity.ovpl -reloadAuthConfig
```

Validating Certificates Using Online Certificate Status Protocol (OCSP)

NNMi supports Online Certificate Status Protocol (OCSP) to check for revoked certificates interactively.

PKI user authentication uses OCSP to verify the revocation status of a certificate by querying an OCSP responder. An OCSP responder provides immediate and accurate revocation information on specific certificates as follows:

- An OCSP client submits a certificate status request to an OCSP responder.
- The OCSP client suspends acceptance of the certificate in question until the OCSP responder provides a digitally signed response.

- The OCSP responder indicates the status of the certificate by returning one of the following values:
 - Good (pass; user is granted access)
 - Revoked (fail; user is denied access)
 - Unknown (fail; user is denied access)

Because the OCSP responder is queried for every certificate, whereas the CRL is downloaded periodically (for example, once per day), OCSP responses might be more up-to-date than corresponding CRLs.

Note: NNMi stores the OCSP configuration in the following location:

- *Windows:* %NnmDataDir%\nmsas\NNM\conf\nms-auth-config.xml
- *Linux:* \$NnmDataDir/nmsas/NNM/conf/nms-auth-config.xml

A default version of the configuration file can be used for reference purposes to view new available options. The default configuration file is stored in the following location:

- *Windows:* %NnmInstallDir%\newconfig\HPOvNnmAS\nmsas\conf\nms-auth-config.xml
- *Linux:* \$NnmInstallDir/newconfig/HPOvNnmAS/nmsas/conf/nms-auth-config.xml

Enabling and Disabling OCSP Checking

To configure OCSP checking, follow these steps:

1. Edit the following file:

Windows: %NnmDataDir%\nmsas\NNM\conf\nms-auth-config.xml

Linux: \$NnmDataDir/nmsas/NNM/conf/nms-auth-config.xml

2. Within the <ocsp> section of the file (find the <ocsp> tag), search for the line that begins with the following text:

```
<enabled>
```

3. Do one of the following:

- To enable OCSP checking, change the line to read as follows:

```
<enabled>>true</enabled>
```

- To disable OCSP checking, change the line to read as follows:

```
<enabled>>false</enabled>
```

4. Save the nms-auth-config.xml file.
5. Run the following command for the change to take effect:

```
nnmsecurity.ovpl -reloadAuthConfig
```

Changing the OCSP Enforcement Mode

By default, NNMi is set to enforce OCSP.

To change the product's enforcement of OCSP, follow these steps:

1. Edit the following file:

Windows: %NnmDataDir%\nmsas\NNM\conf\nms-auth-config.xml

Linux: \$NnmDataDir/nmsas/NNM/conf/nms-auth-config.xml

2. Within the <ocsp> section of the file (find the <ocsp> tag), search for the line that begins with the following text:

```
<mode>
```

3. Change the line to read as one of the following:

```
<mode><value></mode>
```

where <value> is one of the following:

- ENFORCE: Enforce OCSP where specified in the certificates
- ATTEMPT: Check OCSP but allow access if OCSP is not available
- REQUIRE: Require and enforce OCSP in certificates

4. Save the nms-auth-config.xml file.
5. Run the following command for the change to take effect:

```
nnmsecurity.ovpl -reloadAuthConfig
```

Enabling Nonce

For added security (to avoid replay attacks), an OCSP requester can add a nonce to the certificate validation request. A nonce is a random number, attached to each request, that alters the encryption. When the nonce feature is enabled, the OCSP responder computes an appropriate response using the nonce value.

Note: Using a nonce puts more load on the OCSP responder because it cannot precalculate or cache responses. Some OCSP responders may not accept requests with a nonce.

Note: The nonce feature is disabled by default.

To enable the OCSP nonce feature, follow these steps:

1. Edit the following file:

Windows: %NnmDataDir%\nmsas\NNM\conf\nms-auth-config.xml

Linux: \$NnmDataDir/nmsas/NNM/conf/nms-auth-config.xml

2. Within the <ocsp> section of the file (find the <ocsp> tag), search for the line that begins with the following text:

```
<nonce>
```

3. Do one of the following:

- To enable the nonce feature, change the line to read as follows:

```
<nonce>true</nonce>
```

- To disable the nonce feature (and use a general request), change the line to read as follows:

```
<nonce>>false</nonce>
```

4. Save the `nms-auth-config.xml` file.
5. Run the following command for the change to take effect:

```
nmsecurity.ovpl -reloadAuthConfig
```

Specifying the URL of the OCSP Responder

Optionally, you can specify the URL of the OCSP responder as follows:

1. Edit the following file:

```
Windows: %NnmDataDir%\nmsas\NNM\conf\nms-auth-config.xml
```

```
Linux: $NnmDataDir/nmsas/NNM/conf/nms-auth-config.xml
```

2. Within the `<ocsp>` section of the file (find the `<ocsp>` tag), search for the line that begins with the following text:

```
<responder>
```

3. Edit the line to read as follows:

```
<responder><URL></responder>
```

where `<URL>` is the URL associated with the OCSP responder.

4. Save the `nms-auth-config.xml` file.
5. Run the following command for the change to take effect:

```
nmsecurity.ovpl -reloadAuthConfig
```

Note: The OCSP URL must use the HTTP protocol.

- If there is no OCSP URL specified in the `nms-auth-config.xml` file, NNMi attempts to obtain an OCSP responder from the certificate itself.
- If there is no OCSP responder specified in the certificate, NNMi uses the `<mode>` setting to determine what action to take:
 - If the mode is ENFORCE or ATTEMPT, NNMi passes the OCSP validation step for this certificate.
 - If the mode is REQUIRE, NNMi rejects the certificate.

Configuring NNMi to Restrict Certificates Used for NNMi Logon Access

If you are using NNMi with PKI user authentication, you might want to restrict which certificates are considered valid for NNMi logon access.

NNMi supports the following types of restrictions:

- Restrictions on the certificate extended key usage, which can be used to restrict NNMi access to hardware-based certificates or other specific certificates.

- Restrictions on the certificate issuer. These restrictions are intended to prevent a trusted certificate, which is loaded for purposes other than log on purposes, from being used to create log on certificates.

To configure NNMI to restrict certificates used for log on access, do the following:

1. Edit the following file:

Windows: %NnmDataDir%\nmsas\NNM\conf\nms-auth-config.xml

Linux: \$NnmDataDir/nmsas/NNM/conf/nms-auth-config.xml

2. Locate the text block containing the following:

```
<certificateConstraints>
```

3. Use the following examples as a guide to configure NNMI to restrict certificates used for logons (replace values as appropriate):

Example 1: To require client authentication, edit the following section:

```
<!-- client authentication -->
```

```
<extKeyUsage>1.3.6.1.5.5.7.3.2</extKeyUsage>
```

Example 2: To require users to log on using a Microsoft smart card:

```
<!-- Microsoft smart card logon -->
```

```
<extKeyUsage>1.3.6.1.4.1.311.20.2.2</extKeyUsage>
```

Example 3: To accept only certificates signed by a particular CA:

```
<!-- Configures one or more trusted issuers. If this is configured, client certificates must be issued by one of these issuers to be used for client authentication -->
```

```
<trustedIssuer>CN=MyIssuer, OU=MyOrgUnit, O=MyOrg, ST=CO, C=US</trustedIssuer>
```

Note: When multiple extKeyUsage entries are specified, the certificate must contain all of them (Boolean AND). When multiple trustIssuer entries are specified, only one must be the certificate trust issuer (Boolean OR).

4. Run the following command for the change to take effect:

```
nnmsecurity.ovpl -reloadAuthConfig
```

Example: Configuring NNMI to Require a Smart Card Logon

The following example illustrates how to configure NNMI to use PKI user authentication to require a smart card logon.

Note: This example uses the Mixed user authentication strategy.

This example makes the following assumptions:

- The organization is using smart cards for logging on to NNMI.
- The smart card contains a certificate with an email address in the Subject Alternative Name field.
- The organization uses CRLs to check revocation for all certificates.

To complete the example configuration, follow these steps:

1. In the NNMi console, create a user called `myusername@example.com` with guest privileges.
 - a. From the User Accounts view, create the `myusername@example.com` user.

Tip: On the **User Account** form, be sure to select the **Directory Service Account** check box and leave the **Password** field blank. For more information, see the NNMi help.

- b. From the User Account Mappings view, create a new user account mapping to assign the `myusername@example.com` user to the NNMi Guest Users user group.
2. Edit the following file:

Windows: `%NnmDataDir%\nmsas\NNM\conf\nms-auth-config.xml`

Linux: `$NnmDataDir/nmsas/NNM/conf/nms-auth-config.xml`

3. Search for the following text block:

```
<realm name="console">
<mode>FORM</mode>
</realm>
```

4. To enable X.509 certificate authentication, edit the text to read as follows:

```
<realm name="console">
<mode>X509</mode>
</realm>
```

5. Search for the following text block:

```
<principalMapping>
```

6. In the `<principalMapping>` block, include the following line to extract the first match of type `rfc822Name` from the Subject Alternative Name field of the certificate:

```
<subjectAlternativeName type="rfc822Name" />
```

7. Within the `<cr1>` section of the file (find the `<cr1>` tag), search for the line that begins with the following text:

```
<enabled>
```

8. To enable CRL checking, change the line to read as follows:

```
<enabled>>true</enabled>
```

9. Within the `<cr1>` section of the file, locate the text block containing the following text:

```
<mode>
```

10. To require and enforce CRLs, change the line to read as follows:

```
<mode>REQUIRE</mode>
```

11. Locate the text block containing the following:

```
<certificateConstraints>
```

12. To require client authentication, edit the following section:

```
<!-- client authentication -->
<extKeyUsage>1.3.6.1.5.5.7.3.2</extKeyUsage>
```

13. To require users to log on using a Microsoft smart card, add the following lines:

```
<!-- Microsoft smart card logon -->
<extKeyUsage>1.3.6.1.4.1.311.20.2.2</extKeyUsage>
```

14. Save your changes to the `nms-auth-config.xml` file.

15. Change to the directory on the NNMi management server that contains the `nnm-trust.p12` files:

Windows: `%NnmDataDir%\shared\nnm\certificates`

Linux: `$NnmDataDir/shared/nnm/certificates`

16. Import your trusted CA certificate into the `nnm-trust.p12` file. Suppose the `example_ca.cer` file contains the certificate you must use. Run the following command to import the CA certificate into the NNMi `nnm-trust.p12` file:

Windows: `%NnmInstallDir%\bin\nmkeytool.ovpl -import -alias myca -storetype PKCS12 -keystore nnm-trust.p12 -file example_ca.cer`

Linux: `$NnmInstallDir/bin/nmkeytool.ovpl -import -alias myca -storetype PKCS12 -keystore nnm-trust.p12 -file example_ca.cer`

17. Ensure that the user account's name matches the user name contained in the certificate (myusername).

18. Restart the NNMi services:

- Run the `ovstop` command on the NNMi management server.
- Run the `ovstart` command on the NNMi management server.

NNMi is now configured to require a smart card logon.

The following text is similar to how the `nms-auth-config.xml` file might appear after making the configuration changes described in this example:

```
<methods>
  <X509>
    <principalMapping>
      <subjectAlternativeName type="rfc822Name" />
    </principalMapping>
    <certificateConstraints>
      <extKeyUsage>1.3.6.1.5.5.7.3.2</extKeyUsage>
      <extKeyUsage>1.3.6.1.4.1.311.20.2.2</extKeyUsage>
      <trustedIssuer>CN=MyIssuer, OU=MyOrgUnit, O=MyOrg, ST=CO, C=US</trustedIssuer>
    </certificateConstraints>
    <revocation>
      <ordering>CRL OCSP</ordering>
      <mode>CHECK_ALL</mode>
```

```

    </revocation>
    <crl>
      <enabled>>true</enabled>
      <mode>REQUIRE</mode>
      <!-- refresh CRLs every 12 hours -->
      <refreshPeriod>12h</refreshPeriod>
      <!-- remove CRLs that have not been used for 36 hours -->
      <maxIdleTime>36h</maxIdleTime>
    </crl>
    <ocsp>
      <enabled>>false</enabled>
      <mode>ENFORCE</mode>
      <nonce>>false</nonce>
    </ocsp>
  </X509>
</methods>
<realms>
  <realm name="console">
    <mode>X509</mode>
  </realm>
</realms>

```

Configuring CLI Authentication for PKI User Authentication

Authorized users can use the NNMI command line interface (CLI) to configure NNMI settings without having to navigate the NNMI console.

Public Key Infrastructure (PKI) user authentications depend on client-side operating system and web browser settings to perform user authentication. Therefore, CLI sessions cannot use PKI user authentication because the commands run outside the web browser environment. To enable CLI authentication as a non-root user, you can provide authorized users read access to the following file (root users already have read access to this file):

Windows: %NnmDataDir%\nmsas\NNM\conf\props\nms-users.properties

Linux: \$NnmDataDir/nmsas/NNM/conf/props/nms-users.properties

This file contains the encrypted password for the NNMI “system” user. Any user who can read this file can invoke CLI commands as the “system” user.

Note: Windows users who log on as a member of the Administrators group already have read access to

the `nms-users.properties` file, so no further configuration is necessary for Windows users who belong to the Administrators group. For more information about configuring security, see the NNMI help .

Read access to the `nms-users.properties` file can be achieved using the normal Linux `chmod` command. However, it is recommended to configure operating system-based Access Control Lists (ACLs) to provide fine-grained access control to this file. For more information, see ["Setting ACLs to Enable Non-Root Users to Run CLI Commands" below](#).

Setting ACLs to Enable Non-Root Users to Run CLI Commands

ACL commands differ widely among operating systems and file system types on the same operating system. In addition, you might need to configure the operating system to enable ACLs; for example, adding a `,acl` entry to `/etc/fstab` on Linux.

This section provides an example using Linux (RHEL and SuSE) ACL commands with `ext3` and `ext4` file systems. If you are using a different file system type or operating system, see your operating system ACL documentation for more information.

This example gives the operating system user `user1` read permission for the `nms-users.properties` file.

Note: When setting ACL permissions, specify the complete set of permissions for the given file. The provided permissions overwrite the previous permissions.

Grant permission

1. Query the current ACLs using the following command:

```
chacl -l nms-users.properties
```

The output will look something like the following:

```
nms-users.properties [u::rw-,u:user2:r--,u:user3:r--,g::r--,m::r--,o::---]
```

2. Append the new permission (`,u:user1:r--`) to the list output in the square brackets ([]), and run the following command:

```
chacl <results from within square brackets in the ACL list>,u:user1:r-- nms-users.properties
```

Note: ACLs provide user-level control, group-level control, or both. You could also create a Linux group; for example, `nnmiadm`, and then provide read access to the `nms-users.properties` file to the group. Then, by adding or removing Linux users to or from that group, you are also granting or removing access to the `nms-users.properties` file, thereby granting or removing authentication as "system" user to CLI commands.

Caution: Use caution when setting ACLs because incorrect settings that prevent permissions for the `nmsproc` user or `nmsgpr` group can cause NNMI to stop functioning.

List ACLs

Run the following command:

```
chacl -l nms-users.properties
```

Remove permission

1. Query the current ACLs using the following command:

```
chacl -l nms-users.properties
```

2. Identify and delete the user that you want to delete (user1): ,u:user1:r--
3. Paste the rest of the ACL listing into the chacl command:

```
chacl <list results minus user1> nms-users.properties
```

Note: Each of the directories in the `nms-users.properties` file path must be accessible. Normally the permission for these folders is very restrictive, preventing access. This path includes the following directories:

- `$NnmDataDir/nmsas`
- `$NnmDataDir/nmsas/NNM`
- `$NnmDataDir/nmsas/NNM/conf`
- `$NnmDataDir/nmsas/NNM/conf/props`

You can use ACLs also on these folders, or regular Linux `chmod` to grant “search” access (in other words, the execute bit, or 0711 mode) to “other”.

Note: Running the `nmrestore.ovpl` command to restore from an NNMi backup, overwrites the existing ACLs. In this case, after restoring NNMi, recreate and apply your ACLs manually using the procedure for adding users to ACLs described earlier in this section.

Note: In an application failover or high availability (HA) environment, you must set ACLs on both nodes manually by logging onto the primary node, running the appropriate ACL commands, and then repeating the process on the secondary node.

Note: In a Global Network Management (GNM) environment, each separate node might have its own ACLs with different users. For example, a user that has CLI access on a regional manager may not have CLI access on the global manager.

Troubleshooting PKI User Authentication Issues

During PKI user authentication, a user might encounter an error. See the following table for a listing of errors and possible causes.

PKI User Authentication Errors and Possible Causes

Error Message	Possible Cause
401 Not Authenticated	Use of HTTP rather than HTTPS. See "Configure NNMi to Require Encryption for Remote Access" on page 204 for more information.
	User does not have a certificate. See "Managing Certificates" on page 243 for more information.
	User certificate is not trusted by a CA in the nnm-trust.p12. See "Managing Certificates" on page 243 for more information.
	User certificate is expired or not yet valid. See "Managing Certificates" on page 243 for more information.
	User certificate has been revoked or revocation check failed. See "Managing Certificates" on page 243 for more information.
	User certificate failed a constraint check. See "Configuring NNMi to Restrict Certificates Used for NNMi Logon Access" on page 299 for more information.
403 Not Authorized	Mapped user name does not exist in NNMi or the LDAP directory service. See "Configuring NNMi for PKI User Authentication (X.509 Certificate Authentication)" on page 287 for more information.
	Certificate principal to user name mapping is incorrect. See "Configuring NNMi for PKI User Authentication (X.509 Certificate Authentication)" on page 287 for more information.
	User is not in a user group that provides access to the NNMi console. See Configuring Security in the NNMi help for more information.

Note: To troubleshoot, disable HTTP access and turn on logging to help identify issues.

Configuring the Telnet and SSH Protocols for Use by NNMi


The **Actions > 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). The **Actions > Secure Shell... (from client)** menu item invokes the secure shell (SSH) command to the selected node (from the web browser in which the NNMi console is currently running). By default, neither Microsoft Internet Explorer nor Mozilla Firefox defines the telnet command nor the SSH command, so using either of these menu items produces an error message.

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.

Disable the Telnet or SSH Menu Item

If the NNMi users in your deployment environment do not require telnet or SSH connections from the NNMi console, you can disable the respective menu item to remove it from the NNMi console.

Disabling a menu item in the NNMi console applies to all users who log on to the NNMi console on this NNMi management server. To disable the **Telnet** or **Secure Shell** menu item, follow these steps:

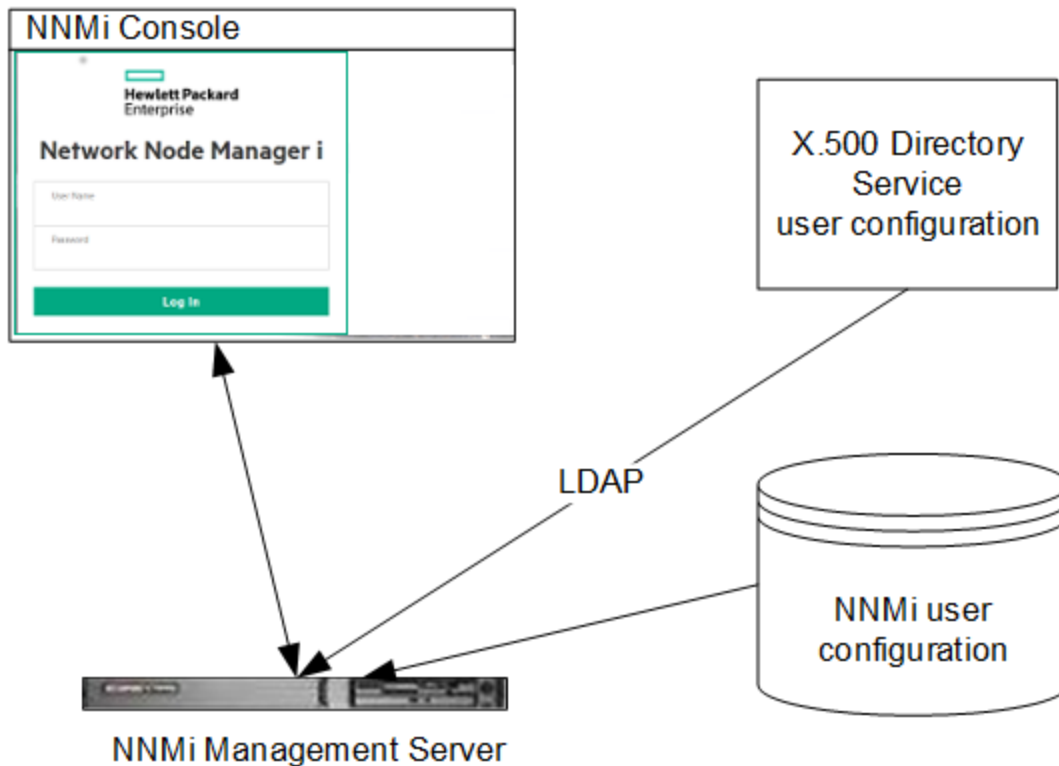
1. In the **Configuration** workspace, expand **User Interface**, and then elect **Menu Items**.
2. In the **Menu Items** view, select the **Telnet... (from Client)** row or the **Secure Shell... (from client)** row, and then click the  **Open** icon.
3. On the **Menu Item** form, clear the **Enabled** 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 more information, see *Control the Actions Menu* in the NNMi help.

Configure a Telnet or SSH Client for the Browser

Configure the browser of the system from where you want to access the NNMi console to use a Telnet or SSH client. For more information, see your web browser's documentation.

Integrating NNMi with a Directory Service through LDAP



This chapter contains information about integrating NNMi with a directory service for consolidating the storage of user names, passwords, and, optionally, NNMi user group assignments. It contains the following topics:

- ["NNMi User Access Information and Configuration Options" below](#)
- ["Configuring NNMi to Access a Directory Service" on page 312](#)
- ["Directory Service Queries" on page 323](#)
- ["Directory Service Configuration for Storing NNMi User Groups" on page 334](#)
- ["Verify the Directory Service Configuration" on page 334](#)
- ["LDAP Configuration File Reference" on page 335](#)

NNMi User Access Information and Configuration Options

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 command line.

- **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 of NNMi commands to the user.

NNMi provides several options for where the NNMi user access information is stored, as described in the following topics. The following table indicates the databases that store the NNMi user access information for each configuration option.

Note: If a user is not specified using External (Option 3), NNMi does not have a mechanism for enforcing password policies, such as password strength checks and other account protection mechanisms. It is recommended that you implement best practices for password policy management, including requiring that users change passwords at regular intervals.

Options for Storing User Information

Mode	User Accounts	User Group	User Group Membership
Internal (Option 1)	NNMi	NNMi	NNMi
Mixed (Option 2)	Mixed (account name in NNMi, account passwords in LDAP)	NNMi	NNMi
External (Option 3)	Directory Service	Both	Directory Service

NNMi communicates with the directory service using Lightweight Directory Access Protocol (LDAP). If you want to use LDAP with NNMi, use one of the following modes shown in previous table:

- **Mixed Mode (Originally Referred to as Option 2):** Some NNMi User Information in the NNMi Database and Some NNMi User Information in the Directory Service
Using mixed mode involves configuring NNMi to store user names, user groups and user group mappings in the NNMi database, and relying on the directory service for user names and passwords (User Accounts). That means that account name information must be stored in both NNMi and LDAP, however account passwords should only be stored in LDAP.
- **External Mode (Originally Referred to as Option 3):** All NNMi User Information in the Directory Service
When using external mode, there is no need to add user account information to NNMi, as all user account information is stored using LDAP.

NNMi's LDAP configuration file: In both the modes, NNMi retrieves the LDAP server information from a configuration file. You can use the `ldap.properties` or `nms-auth-config.xml` file to specify the details of the LDAP server information.

When adding new user accounts, or modifying existing accounts using mixed mode, you must select the **Directory Service Account** check box. When configuring User Accounts do not select the **Directory Service Account** check box for some users and not select it for others as a method of combining internal, mixed, and external modes. Doing so is an unsupported configuration.

When NNMi is integrated with a directory service for some or all of the user access information, the user account and user group definition statement on the **Server** tab of the **System Information** window indicates the type of information that was obtained through LDAP queries.

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

Internal Mode (Originally Referred to as Option 1): All NNMi User Information in the NNMi Database

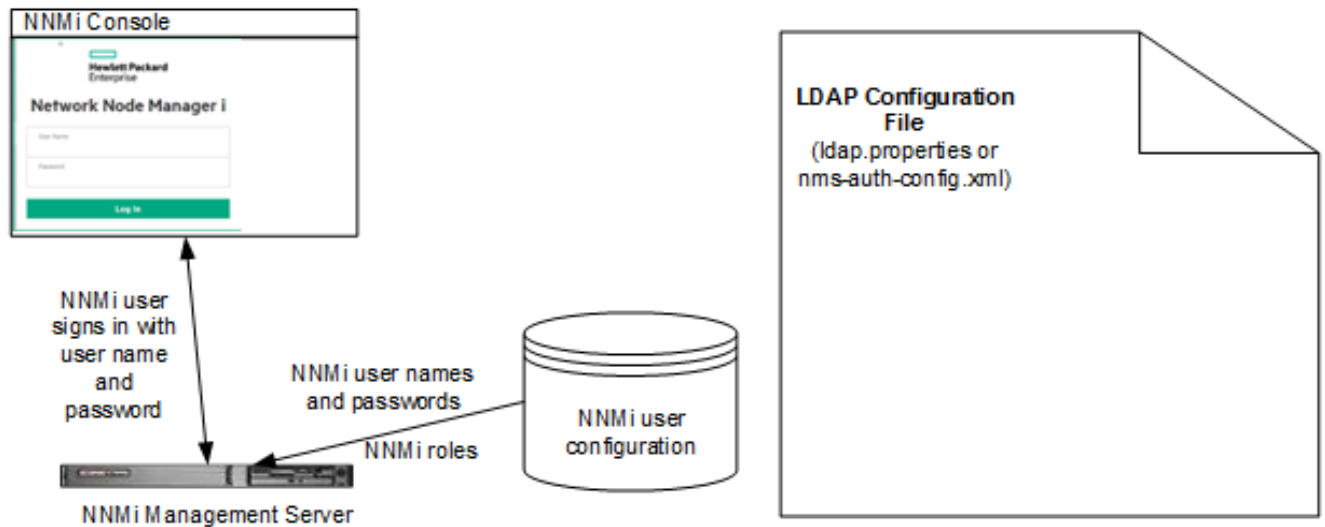
With configuration using the internal mode, NNMi accesses the NNMi database for all user access information, which the NNMi administrator defines and maintains in the NNMi console. The user access information is local to NNMi. NNMi does not access a directory service, and NNMi is not configured to retrieve information from the LDAP configuration file.

The following diagram 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 available.

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

NNMi User Sign-in Information Flow for the Internal Mode



Mixed Mode (Originally Referred to as Option 2): Some NNMi User Information in the NNMi Database and Some NNMi User Information in the Directory Service

With configuration using the mixed mode, 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 password 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 the directory service structure for user names to NNMi.

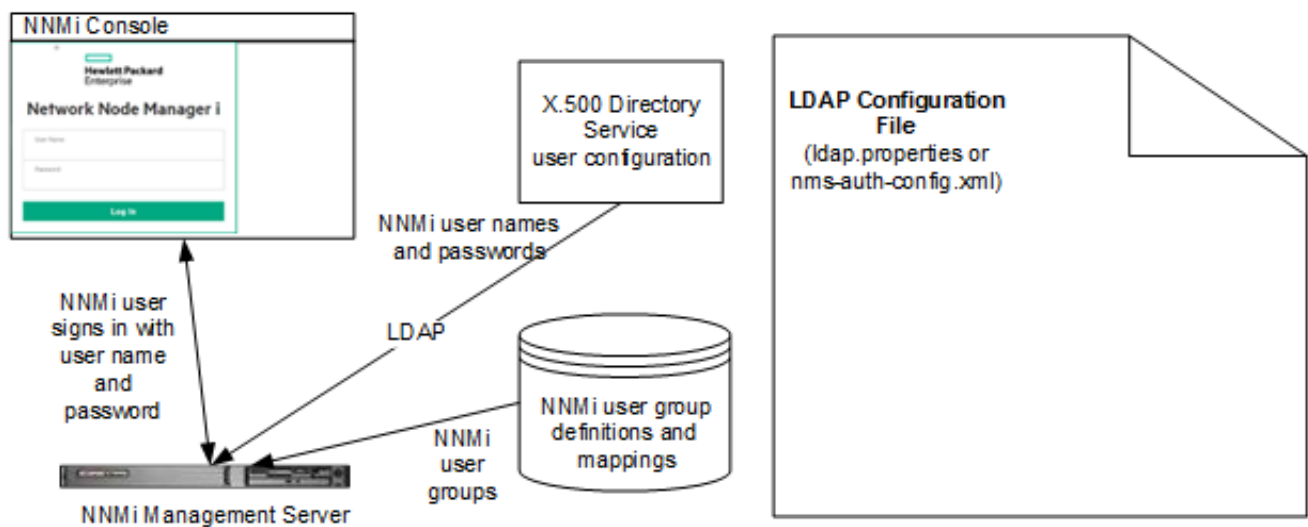
Because user names must be entered in two places, user name maintenance must be performed in both places.

The following diagram shows the information flow for this option, 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 control the user groups instead of requiring a directory service change for each user group change.
- The directory service group definitions are available.

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

NNMi User Sign-in Information Flow for Using Mixed Mode



External Mode (Originally Referred to as Option 3): All NNMi User Information in the Directory Service

With configuration using the external mode, NNMi accesses a directory service for all user access information, which is defined externally to NNMi and is available to other applications. Membership in one or more directory service groups determines the NNMi user groups for the user.

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 the directory service database schema for user names and groups to NNMi.

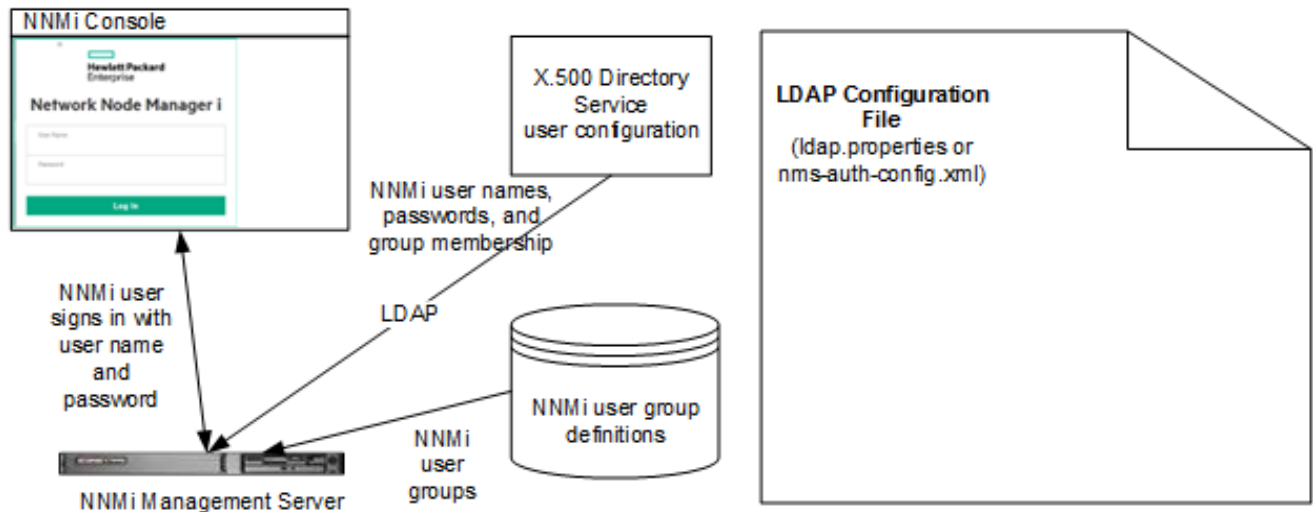
The following diagram shows the information flow for this option, which is appropriate for environments where the directory service can be modified to include user groups that align with the people who need access to NNMi.

Because this option is an expansion of the mixed mode scenario, HPE recommends 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 information about integrating with a directory service for all user information, see the rest of this chapter and *Control Access with a Directory Service* in the NNMi help.

NNMi User Sign-in Information Flow for Using External Mode



Configuring NNMi to Access a Directory Service

You can configure directory service access in *one* of the following files:

- **nms-auth-config.xml**

Note: HPE recommends that the `nms-auth-config.xml` file be used for new configurations.

The file is located at:

- *Windows:* %nmdataDir%\nmsas\NNM\conf
- *Linux:* \$NmDataDir/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:* %nminstalldir%\newconfig\HPOvNmAS\nmsas\conf
- *Linux:* \$NnmInstallDir/newconfig/HPOvNmAS/nmsas/conf

Tip: You can also copy the entire <ldapLogin> element from the sample nms-auth-config.xml file, and then make necessary modifications.

- **ldap.properties**

Note: The ldap.properties file is now deprecated. HPE recommends that the nms-auth-config.xml file be used for new configurations. You cannot configure NNMi to work with multiple LDAP servers in different domains if you use the ldap.properties file.

The file is located at:

- *Windows:* %NNM_SHARED_CONF%\ldap.properties
- *Linux:* \$NNM_SHARED_CONF/ldap.properties

Note: You cannot use both the nms-auth-config.xml and ldap.properties files at the same time.

For information about this file, see ["LDAP Configuration File Reference" on page 335](#). Also see ["Examples" on page 342](#).

For information about the general structure of a directory service, see ["Directory Service Queries" on page 323](#).

For configuration with mixed mode, complete the following tasks:

- [Task 1: Back up the Current NNMi User Information](#)
- [Task 2: Optional. Configure Secure Communications to the Directory Service](#)
- [Task 3: Configure User Access from the Directory Service](#)
- [Task 4: Test the User Name and Password Configuration](#)
- [Task 9: Clean up to Prevent Unexpected Access to NNMi](#)
- ["Task 10: Optional. Map the User Groups to Security Groups" on page 323](#)

For configuration with external mode, complete the following tasks:

- [Task 1 Back up the Current NNMi User Information](#)
- [Task 2: Optional. Configure Secure Communications to the Directory Service](#)
- [Task 3: Configure User Access from the Directory Service](#)
- [Task 4: Test the User Name and Password Configuration](#)
- [Task 5: \(Configuration Option 3 only\) 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 more information, see ["Directory Service Configuration for Storing NNMi User Groups" on page 334](#).

- [Task 6: \(Configuration Option 3 only\) Map the Directory Service Groups to NNMi User Groups](#)

- [Task 7: \(Configuration Option 3 only\) Test the NNMi User Group Configuration](#)
- [Task 8: \(Configuration Option 3 only\) Verify NNMi User Groups for Incident Assignment](#)
- [Task 9: Clean up to Prevent Unexpected Access to NNMi](#)
- [Task 10: Optional. Map the User Groups to Security Groups](#)

Task 1 Back up the Current NNMi User Information

Back up the user information in the NNMi database:

```
nmmconfigexport.ovpl -c account -u <user>
-p <password> -f NNMi_database_accounts.xml
```

Task 2 Optional. Configure Secure Communications to the Directory Service

If the directory service requires the use of secure sockets layer (SSL), import your company's certificate into the NNMi truststore as described in ["Configuring an SSL Connection to the Directory Service" on page 262](#).

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. This task includes the following sections:

Note: Do one of the following depending on your environment or configuration choice.

- [Using nms-auth-config.xml](#)
- [Using ldap.properties](#)

(For detailed configuration instructions, see ["User Identification" on page 330](#).)

Using nms-auth-config.xml

Use the `nms-auth-config.xml` file when you want to configure multiple LDAP servers (in a federated LDAP environment or when the LDAP servers are in an HA cluster).

1. Go to the following directory:
 - *Windows:* %nmdatadir%\nmsas\NNM\conf
 - *Linux:* \$NmDataDir/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:

Tip:

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:* %nminstalldir%\newconfig\HPOvNnmAS\nmsas\conf
- *Linux:* \$NnmInstallDir/newconfig/HPOvNnmAS/nmsas/conf

Tip: You can also copy the entire <ldapLogin> element from the sample nms-auth-config.xml file, and then make necessary modifications.

Elements of the ldapLogin Section of nms-auth-config.xml

<pre><enabled> </enabled></pre>	<p>Specify true to use the nms-auth-config.xml file. By default, this element is set to false.</p>
<pre><userRoleFilterList> </userRoleFilterList></pre>	<p>Specify the NNMi roles to which NNMi users can assign incidents.</p> <p>To assign incidents to all operators, administrators, and guests, add this:</p> <pre><userRoleFilterList> admin guest level2 level1 </userRoleFilterList></pre>
<pre><connectTimeLimit> </connectTimeLimit></pre>	<p>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></p>
<pre><searchTimeLimit> </searchTimeLimit></pre>	<p>Specify the search timeout value in milliseconds. The default value is 10000 (10 seconds). If you are encountering timeouts during NNMi user sign in, increase this value. For example: <searchTimeLimit>10000</searchTimeLimit></p>
<pre><server></pre>	<p>Container element to contain all LDAP configuration information.</p>
<pre><host> </host></pre>	<p>URL of the LDAP server with port. For example: ldap://hostname.domain.com</p>
<pre><secure> </secure></pre>	<p>Specify true if you want to use HTTPS. Otherwise, specify false.</p>
<pre></server></pre>	

Elements of the ldapLogin Section of nms-auth-config.xml, continued

<p>Note: Repeat the server element when you want to use multiple LDAP servers. Use one server element for each LDAP server.</p>	
<bindCredential>	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.
<users>	Container element to include all user configuration details.
<userSearch>	Container element to include the configuration information for searching users.
<base> </base>	For example: <code><base> SAMAccountName={0} </base></code> <code><base> uid={0} </base></code>
<baseContextDN> </baseContextDN>	For Active Directory, specify the portion of the directory service domain that stores user records. For example: <i>For Active Directory</i> <code>CN=user,OU=Users,OU=Accounts,DC=mycompany,DC=com</code> <i>For other LDAP technologies</i> <code>ou=People,o=example.com</code>
</userSearch> </users>	

Elements of the ldapLogin Section of nms-auth-config.xml, continued

Note: You can repeat the configuration element when you want to use multiple LDAP servers with different LDAP configurations.

For example, if one unit in your organization uses Windows and Active Directory, and another unit uses Linux with OpenLDAP, you could specify two different `<configuration>` elements, one for Active Directory and the other for OpenLDAP.

. Use one set of base-baseContextDN elements for each LDAP server.

In an HA cluster of LDAP servers, the identity information is identical across servers and you should use multiple server elements in a single configuration element, instead of using multiple configuration elements.

However, in other environments, it is possible to specify different base formats for different LDAP servers (for example, SAMAccountName for one and uid for the other).

4. After editing the `nms-auth-config.xml` file (in the `<NnmInstallDir>/nmsas/NNM/conf` directory), run the following command:
 - *Windows:* `%nmminstalldir%\bin\nnmldap.ovpl -reload`
 - *Linux:* `$NnmInstallDir/bin/nnmldap.ovpl -reload`

Using ldap.properties

1. Back up the `ldap.properties` file that was shipped with NNMI, and then open the file in any text editor.
2. Specify the URL for accessing the directory service.
 - a. Uncomment the following line:


```
java.naming.provider.url
```
 - b. Set the property to `ldap://<myldapserver>:<port>/`.

In this instance, `<myldapserver>` is the fully-qualified hostname of the directory server and `<port>` is the communication port of the directory server.

Example:

```
java.naming.provider.url=ldap://testsystem.example.com:636
```

3. Specify the security mode.
 - a. Uncomment the following line:


```
java.naming.security.provider
```
 - b. Set the property to SSL if you want NNMI to communicate with the directory server securely.

Example:

```
java.naming.security.provider=SSL
```

4. If your directory service installation does not support anonymous access, specify credentials for a valid directory service user.

- a. Uncomment the following lines:

```
bindDN
bindCredential
```

- b. Set these properties to the following values:

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

In this instance, *<mydomain>* with the name of the directory server domain; *<myusername>* and *<mypassword>* are the user name and password for accessing the directory server.

Note: If you plan to add the password in plain text, specify a user name with read-only access to the directory service. If you plan to specify an encrypted password, use the following command to encrypt the plain text password before adding it to the `ldap.properties` file:

```
nnmldap.ovpl -encrypt <mypassword>
```

This encrypted password only works for the NNMi instance you create it for. Do not attempt to use it for a different NNMi instance.

For more information see the *nnmldap.ovpl* reference page, or the Linux manpage.

5. Specify the portion of the directory service domain that stores user records.

- a. Uncomment the following line:

```
baseCtxDN
```

- b. Set this properties to the portion of the directory service domain that stores user records.

Examples:

- o Microsoft Active Directory


```
baseCtxDN=CN=Users,DC=hostname,DC=example,
DC=com
```
- o Other LDAP


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

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

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`, add the following content *before* the `usersearch` element:

```
<defaultRoles>
<role>guest</role>
</defaultRoles>
```

- In `ldap.properties`, add `defaultRole=guest`.

2. Save the LDAP configuration file.
3. Force NNMi to re-read the file by running the following command:
`nnmldap.ovpl -reload`
4. Log on to the NNMi console with a user name and password that are defined in the directory service.

Tip: 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, continue with [step 8](#) of this task.
 - If user sign in does not work correctly, continue with [step 6](#), next.

Tip: 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:

```
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. Suggestions include:

- Verify that you completed [Task 3](#) correctly.
 - Follow the detailed configuration process in "[User Identification](#)" on page 330.
7. Repeat [step 1](#) through [step 5](#) until you see the expected result when signing in to the NNMi console.
 8. After you can log on, choose your strategy:
 - If you plan to store NNMi user group membership in the NNMi database (configuration using mixed mode), continue with [Task 9](#).
 - If you plan to store NNMi user group membership in the directory service (configuration using external mode), continue with [Task 5](#), next.

Task 5: (External Mode only) Configure Group Retrieval from the Directory Service

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

- [Using the nms-auth-config.xml File](#)
- [Using ldap.properties](#)

Note: Do one of the following depending on your environment or configuration choice.

(For detailed configuration instructions, see "[User Group Identification](#)" on page 331.)

Using the nms-auth-config.xml File

- Go to the following directory:
 - Windows:* %nnmdatadir%\nmsas\NNM\conf
 - Linux:* \$NnmDataDir/nmsas/NNM/conf
- Take a backup of the nms-auth-config.xml file, and then open the file with a text editor.
- Modify the following elements:

Tip:

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:* %nnminstallldir%\newconfig\HPOvNnmAS\nmsas\conf
- Linux:* \$NnmInstallDir/newconfig/HPOvNnmAS/nmsas/conf

Tip: You can also copy the entire <ldapLogin> element from the sample nms-auth-config.xml file, and then make necessary modifications.

Elements of the ldapLogin Section of nms-auth-config.xml

<roleSearch>	Placeholder element to include the user role information.
<roleBase> <i>member</i> = {1}</roleBase>	Replace <i>member</i> 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: <ul style="list-style-type: none"> <i>For Microsoft Active Directory</i> CN=Users,DC= Ldapserver,DC=mycompany,DC=com <i>For other LDAP technologies</i> ou=Groups,o=example.com
</roleSearch>	

4. Save the file.
5. Run the following command:
nnmldap.ovpl -reload

Using ldap.properties


1. Back up the `ldap.properties` file, and then open the file in any text editor.
2. Uncomment the `rolesCtxDN` property.
3. Set the property to the portion of the directory service domain that stores group records.

Examples:

- Microsoft Active Directory
`rolesCtxDN=CN=Users,DC=hostname,DC=example,DC=com`
- Other LDAP
`rolesCtxDN=ou=Groups,o=example.com`


4. Save the file.
5. Run the following command:
nnmldap.ovpl -reload

Task 6: (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 the  **Save and Close** icon.
 - e. Repeat **step b** through **step d** for each of the **guest**, **level1**, and **level2** rows.

Tip: 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 the *** New** icon, and then enter the information for the group:

- o Set **Unique Name** to any unique value. Short names are recommended.
 - o Set **Display Name** to the value users should see.
 - o Set **Directory Service Name** to the full distinguished name of the directory service group.
 - o 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.

Tip: These mappings provide topology object access in the NNMI console. Each directory service group can be mapped to multiple NNMI user groups.

Task 7: (External Mode only) Test the NNMI User Group Configuration

1. Save NNMI's LDAP configuration file (`ldap.properties` or `nms-auth-config.xml`).
2. Force NNMI to re-read the LDAP configuration file by running the following command:
`nnmldap.ovpl -reload`
3. Log on to the NNMI console with a user name and password that are defined in the directory service.

Note: 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 Group** view) in the title bar of the NNMI console.
 - If user signin works correctly, continue with [Task 8](#).
 - If user signin does not work correctly, continue with [step 5](#), next.

Tip: 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. Suggestions include:
 - Verify that you completed [Task 5](#) correctly.
 - Verify that you completed [Task 6](#) correctly for each of the predefined NNMI user groups.
 - Follow the detailed configuration process in "[User Group Identification](#)" on [page 331](#).
6. Repeat [step 1](#) through [step 4](#) until you see the expected result when signing in to the NNMI console.

Task 8: (External Mode only) Verify NNMi User Groups for Incident Assignment

1. Log on 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.

Task 9: Clean up to Prevent Unexpected Access to NNMi

1. Optional. Change the value of, or comment out, the `defaultRole` element or parameter in the LDAP configuration file.
2. (Mixed Mode only) To store user group membership in the NNMidatabase, reset the user access information in the NNMidatabase as follows:
 - a. Remove any pre-existing user access information. (Delete all rows in the **User Accounts** view.)
For instructions, see *Delete a User Account* in the NNMi help.
 - b. For each NNMi user, create a new object in the **User Accounts** view for the user name.
 - o For the **Name** field, enter the user name as defined in the directory service.
 - o Select the **Directory Service Account** check box.
 - o Do not specify a password.
 For more information, see *User Account Tasks* in the NNMi help.
 - c. For each NNMi user, map the user account to one or more NNMi user groups.
For instructions, see *User Account Mapping Tasks* in the NNMi help.
 - d. Update incident ownership so that each assigned incident is associated with a valid user name.
For instructions, see *Manage Incident Assignments* in the NNMi help.
3. (External Mode only) To rely on the user group membership in the directory service, 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 instructions, see *Delete a User Account* in the NNMi help.
 - b. Update incident ownership so that each assigned incident is associated with a valid user name.
For instructions, see *Manage Incident Assignments* in the NNMi help.

Task 10: Optional. Map the User Groups to Security Groups

For instructions, see *Security Group Mapping Tasks* in the NNMi help.

Directory Service Queries

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

This section contains the following topics:

- ["Directory Service Access" on the next page](#)
- ["Directory Service Content" on the next page](#)

- ["Information Owned by the Directory Service Administrator" on page 328](#)
- ["User Identification" on page 330](#)
- ["User Group Identification" on page 331](#)

Directory Service Access

LDAP queries to a directory service use the following format:

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

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

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

Tip: 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.

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.

Example content structure for Active Directory

In this example, the following items are of interest:

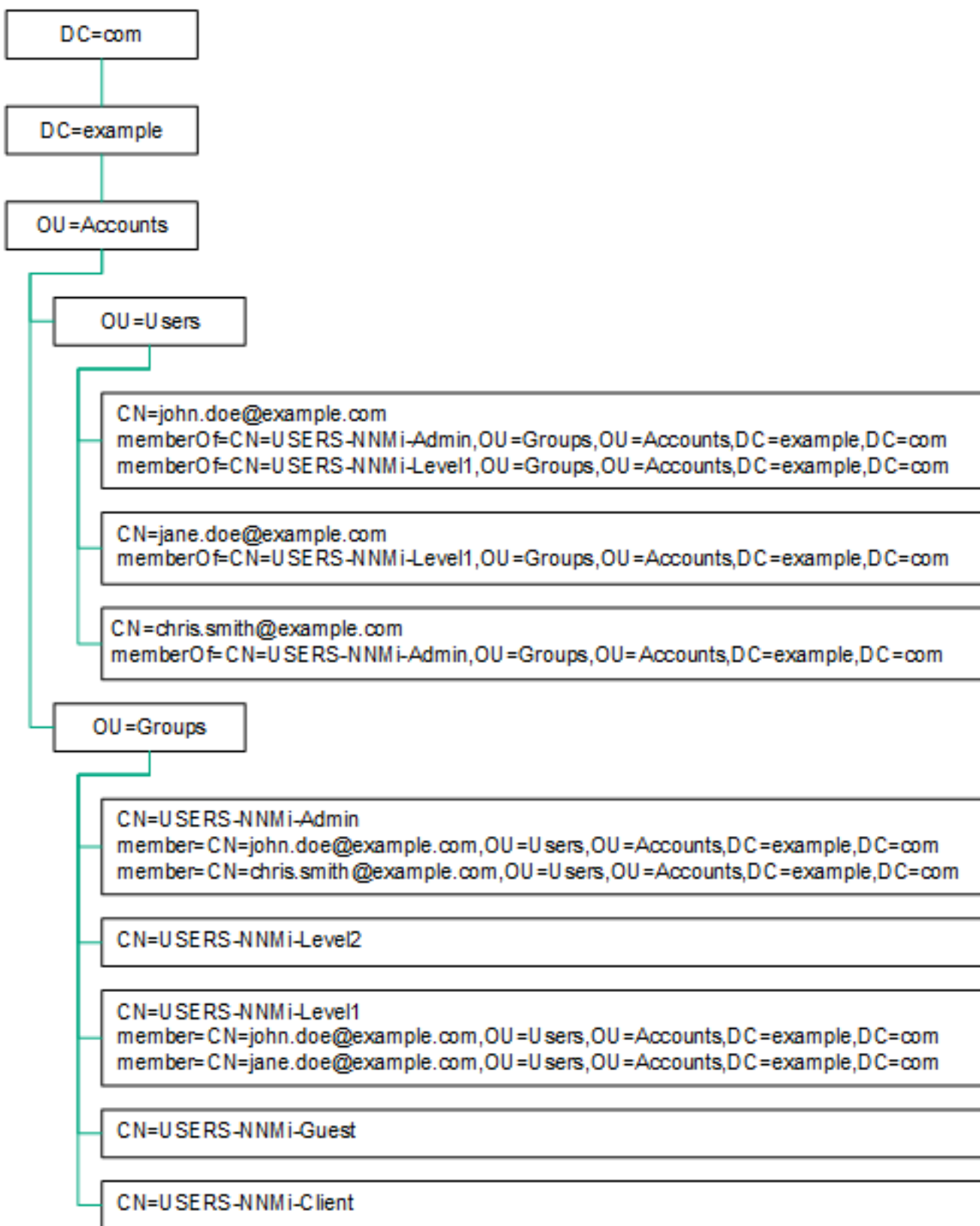
- The distinguished name of the user John Doe is:
CN=john.doe@example.com,OU=Users,OU=Accounts,DC=example,DC=com
- The distinguished name of the group USERS-NNMi-Admin is:
CN=USERS-NNMi-Admin,OU=Groups,OU=Accounts,DC=example,DC=com
- The group attribute that stores the directory service user ID is:
member

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 diagram illustrates this directory service domain.

Example Domain for Active Directory



Example content structure for other directory services

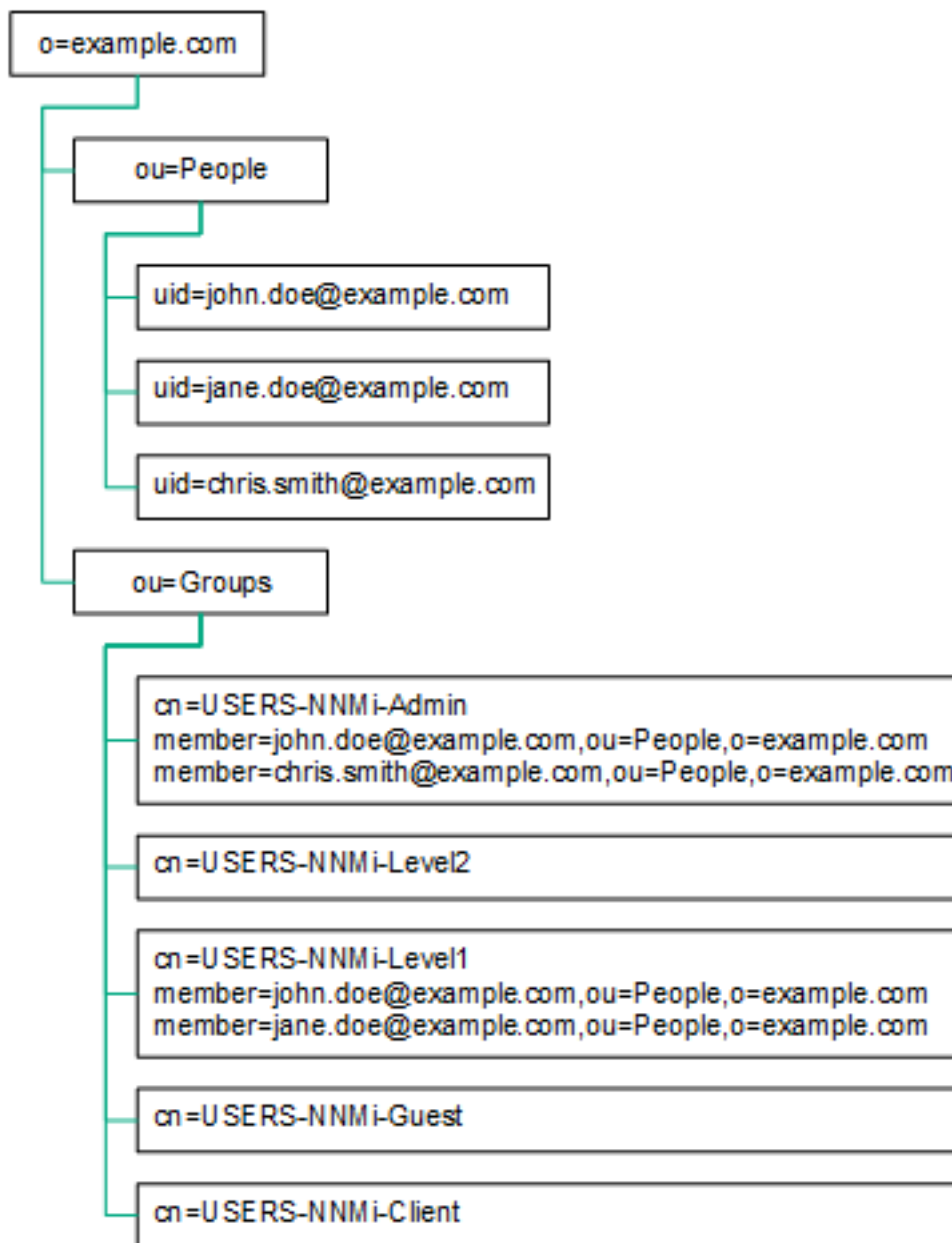
In this example, the following items are of interest:

- 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
description: Group of users for NNMi administration.
member: uid=john.doe@example.com,ou=People,o=example.com
member: uid=chris.smith@example.com,ou=People,o=example.com
```

Example Domain for Other Directory Services



Information Owned by the Directory Service Administrator

The following tables 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 (mixed mode only), gather the information for [Retrieving User Names and Passwords from a Directory Service](#).
- If you plan to use the directory service for all NNMi access information (external mode only), gather the information for each of the following tables.

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	<ul style="list-style-type: none"> • 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 "Configuring an SSL Connection to the Directory Service" on page 262.	
The distinguished name for one user name that is stored in the directory service (to demonstrate the directory service domain)	CN=john.doe@example.com, OU=Users,OU=Accounts, DC=example,DC=com	uid=john.doe@example.com, ou=People,o=example.com

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.	<ul style="list-style-type: none"> • ou=Groups,o=example.com • cn=USERS-NNMi-*, ou=Groups,o=example.com
The method of identifying a user within a group	<ul style="list-style-type: none"> • CN=john.doe@example.com, OU=Users,OU=Accounts, DC=example,DC=com • CN=john.doe@example.com 	<ul style="list-style-type: none"> • 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	<ul style="list-style-type: none"> • 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 	<ul style="list-style-type: none"> • 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

Information for Retrieving Group Membership from a Directory Service, continued

Information	Active Directory Example	Other Directory Services Example
	<ul style="list-style-type: none"> • CN=USERS-NNMi-Guest, OU=Groups,OU=Accounts, DC=example,DC=com 	

User Identification

User identification applies to mixed mode and 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.

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 (the `baseFilter` value and the `baseCtxDN` value in the `ldap.properties` 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.

For 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 to which this user belongs in the NNMi console.

For 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 to which this user belongs in the directory service (as mapped to NNMi user groups in the NNMi console).

Active Directory user identification example

In the `nms-auth-config.xml` file

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
```

In the `Ldap.properties` file

If `baseFilter` is set to `CN={0}`, `baseCtxDN` is set to `OU=Users,OU=Accounts,DC=example,DC=com`, and a user signs in to NNMi as `john.doe`, 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

In the `nms-auth-config.xml` file

If the `nms-auth-config.xml` file contains `<base>uid={0}@example.com</base><baseContextDN>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
```

In the `Ldap.properties` file

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

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

User Group Identification

User group identification applies to 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.

NNMi User Group Name Mappings

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

Note: The NNMi Global Operators User Group (`globalops`) grants access to all topology objects only. A

user must be assigned to one of the other User Groups (admin, level2, level1, or guest) to access the NNMi console.

The administrator should not map the globalops User Group to any security group because this User Group is, by default, mapped to all security groups.

Configuring User Group Retrieval from the Directory Service (Detailed Approach)

If the simple approach described in [Task 5](#) 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 [Determining How the Directory Service Identifies a Group and Group Membership \(LDAP Browser Approach for Active Directory\)](#).
 - *LDAP browser approach for other directory services:* See [Determining How the Directory Service Identifies a Group and Group Membership \(LDAP Browser Approach for Other Directory Services\)](#).
 - *Web browser approach for other directory services:* See [Determining How the Directory Service Identifies a Group \(Web Browser Approach\)](#).
3. Configure the LDAP configuration file.
 - **Using the nms-auth-config.xml file:**
 - i. Open the nms-auth-config.xml file in any text editor.
 - ii. 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).
 - iii. 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 Microsoft Active Directory*
CN=Users,DC=Ldapserver,DC=mycompany,DC=com
 - *For other LDAP technologies*
ou=Groups,o=example.coms
 - **Using the ldap.properties file:**
 - i. Open the ldap.properties file in any text editor.
 - ii. Set the rolesCtxDN parameter to the elements of the distinguished group name that are the same for multiple groups.
 - iii. Set the roleFilter parameter to correlate user names to the way user names are stored for groups in the directory service. Replace the actual user name with one of the following

expressions:

- Use {0} to denote the user name entered for signin (for example, john.doe).
 - Use {1} to denote the distinguished name of the authenticated user as returned by the directory service (for example, uid=john.doe@example.com,ou=People,o=example.com).
- iv. Set the uidAttributeID parameter to the name of the group attribute that stores the user ID.

4. Test the configuration as described in "[Configuring NNMi to Access a Directory Service](#)" on page 312.

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.

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.

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

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


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

 - *<directory_service_host>* is the fully-qualified name of the computer that hosts the directory service.
 - *<port>* is the port that the directory service uses for LDAP communication.
 - *<group_search_string>* is the distinguished name for a group name that is stored in the directory service, for example: cn=USERS-NNMi-Admin,ou=Groups,o=example.com
2. Evaluate the results of the directory service access test.
 - If you see a message that the directory service does not contain the requested entry, verify the value of *<group_search_string>*, and then repeat [step 1](#).
 - If you see the appropriate list of groups, the access information is correct.
3. Examine the group properties to determine the format of the names for the users associated with that group.

Directory Service Configuration for Storing NNMi User Groups

If you plan to store NNMi user groups in the directory service (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.

Verify the Directory Service Configuration

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.properties` file.

2. Force NNMi to re-read the LDAP configuration file by running 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.

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

5. View the log file to verify that the sign-in request is correct, and to determine if any errors occurred:

Windows: %NnmDataDir%\log\nnm\nnm.log

Linux: \$NnmDataDir/log/nnm/nnm.log

- A message similar to the following line indicates that the directory service requires HTTPS communication. In this case, enable SSL as described in "[Configuring an SSL Connection to the Directory Service](#)" on page 262.

```
javax.naming.AuthenticationNotSupportedException: [LDAP: error code 13 - confidentiality required]
```

- A message similar to the following line indicates that a timeout occurred while communicating with the directory service. In this case, increase the value of `searchTimeLimit` in the `ldap.properties` file.

```
javax.naming.TimeoutException: [LDAP: error code 3 - Timelimit Exceeded]
```

LDAP Configuration File Reference

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:* %nmmdatadir%\nmsas\NNM\conf
- *Linux:* \$NmDataDir/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\HPOvNmAS\nmsas\conf
- *Linux:* \$NmInstallDir/newconfig/HPOvNmAS/nmsas/conf

Tip: 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 (in the `<NmInstallDir>/nmsas/NNM/conf` directory), force NNMi to read the LDAP configuration again by running the following command:

- *Windows:* %nnminstallDir%\bin\nnmldap.ovpl -reload
- *Linux:* \$NmInstallDir/bin/nnmldap.ovpl -reload

```
<ldapLogin>
```

```
<!-- This is the on/off switch for LDAP authentication. Set to true to use LDAP-based authentication-->
```

```
    <enabled>true</enabled>
```

```
<!-- This element enables you to specify which users can assign incidents.-->
```

```
    <userRoleFilterList>admin guest level2 level1</userRoleFilterList>
```

```
<!-- If <enabled> is set to true, define one or more <configuration> elements to specify LDAP parameters -->
```

```
    <configuration>
```

```
<!-- The filter (optional) is matched against the user, that tries to log on, to determine if this is the right configuration to use. This is useful when multiple configurations 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>10000</searchTimeLimit>
    <connectTimeLimit>10000</connectTimeLimit>
<!-- Define at least one server URL; multiple servers can be specified for High-Availability clusters.-->
    <server>
        <hostname>ldaps://ldap.domain1.com</hostname>
        <secure>>true</secure>
    </server>
    <server>
        <hostname>ldaps://ldap.domain2.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 encrypted 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 intention is to skip non-
applicable 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 against this configuration -->

```



```

    <defaultRoles>
      <role>guest</role>
    </defaultRoles>

```

<!-- One or more search configuration for locating user accounts. The pattern "{0}" in 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 against 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 authenticated 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>

```

ldap.properties

Note: The `ldap.properties` file is now deprecated.

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

- *Windows:* `%NNM_SHARED_CONF%\ldap.properties`
- *Linux:* `$NNM_SHARED_CONF/ldap.properties`

Note: You cannot configure NNMI to work with multiple LDAP servers in different domains if you use the `ldap.properties` file.

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

- To comment out a line, begin that line with a number sign character (#).
- The following rules apply to special characters:
 - To specify a backslash character (\), comma (,), semicolon (;), plus sign (+), less than sign (<), or greater than sign (>), escape the character with a backslash character. For example: `\\` or `\+`
 - To include a space character () as the *first* or *last* character in a string, escape the space character with a backslash character (\).
 - To include a number sign character (#) as the *first* character in a string, escape the number sign character with a backslash character (\).

Characters not mentioned here do not need to be escaped or quoted.

Note: After editing the `ldap.properties` file, force NNMI to re-read the LDAP configuration by running the following command:

```
nnmlldap.ovpl -reload
```

The following table describes the parameters in the `ldap.properties` file.

Note: The initial `ldap.properties` file might not include all parameters that are listed in the following table. Add the parameters you need.

Parameters in the ldap.properties File

Parameter	Description
<code>java.naming.provider.url</code>	<p>Specifies the URL for accessing the directory service.</p> <p>The format is the protocol (<code>ldap</code>), followed by the fully-qualified host name of the directory server, optionally followed by the port number. For example:</p> <pre>java.naming.provider.url=ldap://ldap.example.com:389/</pre> <p>If the port number is omitted the following defaults apply:</p>

Parameters in the `ldap.properties` File, continued

Parameter	Description
	<ul style="list-style-type: none"> For non-SSL connections, the default port is 389. For SSL connections, the default port is 636. <p>If you specify multiple directory service URLs, NNMI uses the first directory service when possible. If that directory service is not accessible, NNMI queries the next directory service in the list, and so forth. Separate each URL with a single space character. For example:</p> <pre>java.naming.provider.url=ldap://ldap1.example.com/ ldap:// ldap2.example.com/</pre> <p>Configuring this parameter enables LDAP communication between NNMI and the directory service. To disable LDAP communication, comment out this parameter, and then save the file. NNMI ignores the configuration in the <code>ldap.properties</code> file.</p>
<code>java.naming.security.protocol</code>	<p>Specifies the connection protocol specification.</p> <ul style="list-style-type: none"> If the directory service is configured to use LDAP over SSL, set this parameter to <code>ssl</code>. For example: <code>java.naming.security.protocol=ssl</code> If the directory service does not require SSL, leave this parameter commented out. <p>For more information, see "Configuring an SSL Connection to the Directory Service" on page 262.</p>
<code>bindDN</code>	<p>For a directory service (such as Active Directory) that does not permit anonymous access, specify the user name for accessing the directory service.</p> <p>For example:</p> <pre>bindDN=region1\john.doe@example.com</pre> <ul style="list-style-type: none"> If you plan to add the password in plain text, specify a user name with read-only access to the directory service. For example: <code>bindCredential=PasswordForJohnDoe</code> If you plan to specify an encrypted password, use the following command to encrypt the plain text password before adding it to the <code>ldap.properties</code> file: <code>nnmldap.ovpl -encrypt <mypassword></code> For example: <code>bindCredential={ENC}uaF22C+0CF9VozBVYj80Aw==</code> <p>This encrypted password only works for the NNMI instance you create it for. Do not attempt to use it for a different NNMI instance. For more information see the <code>nnmldap.ovpl</code> reference page, or the UNIX manpage.</p>
<code>bindCredential</code>	<p>When <code>bindDN</code> is set, specifies the password for the user name that <code>bindDN</code> identifies. For example:</p>

Parameters in the ldap.properties File, continued

Parameter	Description
	bindCredential=PasswordForJohnDoe
baseCtxDN	<p>Specifies the portion of the directory service domain that stores user records.</p> <p>The format is a comma-separated list of directory service attribute names and values. For example:</p> <ul style="list-style-type: none"> baseCtxDN=CN=Users,DC=ldapserver,DC=example,DC=com baseCtxDN=ou=People,o=example.com <p>For more information, see "User Identification" on page 330.</p>
baseFilter	<p>Specifies the format of user names for signing in to NNMI.</p> <p>The format is the name of the directory service user name attribute and a string that relates the entered user sign-in name to the format of names in the directory service. The user name string contains the expression {0} (to denote the user name entered for sign in) and any other characters that are needed to match the directory service formatting of user names.</p> <ul style="list-style-type: none"> If the user name entered for NNMI sign in is the same as the user name stored in the directory service, the value is the replacement expression. For example: <ul style="list-style-type: none"> baseFilter=CN={0} baseFilter=uid={0} If the user name entered for NNMI sign in is a subset of the user name stored in the directory service, include the additional characters in the value. For example: <ul style="list-style-type: none"> baseFilter=CN={0}@example.com baseFilter=uid={0}@example.com <p>For more information, see "User Identification" on page 330.</p>
defaultRole	<p>Optional. Specifies a default role that applies to any directory service user who signs in to NNMI through LDAP. The value of this parameter applies regardless of where user group mappings are stored (in the NNMI database or in the directory service).</p> <p>If a user is directly configured for a predefined NNMI user group, NNMI grants the user the superset of privileges for the default role and the assigned user group.</p> <p>Valid values are as follows: admin, level2, level1, or guest.</p> <p>Note that although admin is a valid value, you should use caution and consider the implications of making admin a default role.</p> <p>These names are the unique names of the predefined NNMI user group names.</p> <p>For example:</p>

Parameters in the ldap.properties File, continued

Parameter	Description
	<p>defaultRole=guest</p> <p>If commented out or omitted, NNMI does not use a default role.</p>
rolesCtxDN	<p>Specifies 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:</p> <ul style="list-style-type: none"> • rolesCtxDN=CN=Users,DC=ldapserver,DC=example,DC=com • rolesCtxDN=ou=Groups,o=example.com <p>In other directory services (not Active Directory), for a faster search, you can identify one or more directory service groups that contain NNMI user groups. If the group names form a pattern, you can specify a wildcard. For example, if the directory service includes groups named USERS-NNMI-administrators, USERS-NNMI-level10operators, and so forth, you could use a search context similar to:</p> <pre>rolesCtxDN=cn=USERS-NNMI-*,ou=Groups,o=example.com</pre> <p>Configuring this parameter enables directory service queries for NNMI user group assignments through LDAP.</p> <p>To disable directory service queries for NNMI user group assignments through LDAP, comment out this parameter, and then save the file. NNMI ignores the remaining user group-related values in the ldap.properties file.</p> <p>For more information, see "User Group Identification" on page 331.</p>
roleFilter	<p>Specifies the format of group member names in the directory service group definitions.</p> <p>The format is the name of the directory service group attribute for user ID and a string that relates the entered user sign-in name to the format of user IDs in the directory service. The user name string contains one of the following expressions and any other characters that are needed to match the directory service formatting of group member names.</p> <ul style="list-style-type: none"> • The expression {0} denotes the user name entered for sign in (for example, john.doe). An example role filter that matches on the (short) user name entered for sign in is: roleFilter=member={0} • The expression {1} denotes the distinguished name of the authenticated user as returned by the directory service (for example, CN=john.doe@example.com,OU=Users,OU=Accounts,DC=example,DC=com) or uid=john.doe@example.com,ou=People,o=example.com). An example role filter that matches on the (full) authenticated user name is: roleFilter=member={1}

Parameters in the ldap.properties File, continued

Parameter	Description
	For more information, see "User Group Identification" on page 331 .
uidAttributeID	Specifies the group attribute that stores the directory service user ID. For example: uidAttributeID=member For more information, see "User Group Identification" on page 331 .
userRoleFilterList	Optional. Limits the NNMi user groups whose associated users can be assigned incidents in the NNMi console. The user groups in this list apply only to directory service user names authenticated through LDAP. This parameter provides functionality that is not available when NNMi user groups are assigned in the NNMi console and stored in the NNMi database. The format is a semicolon-separated list of the unique names for one or more predefined NNMi user group names. userRoleFilterList=admin;globalops;level2;level1
searchTimeLimit	Optional. Specifies the timeout value in milliseconds. The default value is 10000 (10 seconds). If you are encountering timeouts during NNMi user sign in, increase this value. For example: searchTimeLimit=10000

Examples

Example ldap.properties file for Active Directory

An example ldap.properties file follows for Active Directory:

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

Example ldap.properties file for other directory services

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

```

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

```

Switching to the nms-auth-config.xml File

If you want to move NNMI's LDAP configuration to the `nms-auth-config.xml` file from the `ldap.properties` file, follow these steps:

Tip: The `nms-auth-config.xml` file enables you to configure multiple LDAP servers with NNMI. With the `ldap.properties` file, you cannot configure more than one LDAP server with NNMI.

The `ldap.properties` file is now deprecated.

1. Take a backup of the existing `ldap.properties` file. You can use this copy of the file as a reference while completing the following tasks.
2. Complete the tasks listed in ["Configuring NNMI to Access a Directory Service" on page 312](#). Follow the steps highlighted for the `nms-auth-config.xml` file in this section.
3. Run the following command:
 - *Windows:* `%nnminstalldir%\bin\nnmldap.ovpl -reload`
 - *Linux:* `$NnmInstallDir/bin/nnmldap.ovpl -reload`

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.

To configure NNMI to use a non-default IP address with managed nodes:

1. Log on to the NNMI Management Server.
2. Open the following file with a text editor:
 - a. *On Windows:* `%nnmdatadir%\shared\nnm\conf\props\nms-communication.properties`
 - b. *On 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 property of your choice.

Note: 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.

5. Restart the NNMi processes by running the following commands:

- *On Windows:*
 - i. `%nnminstalldir%\bin\ovstop -c`
 - ii. `%nnminstalldir%\bin\ovstart -c`
- *On Linux:*
 - i. `/opt/OV/bin/ovstop -c`
 - ii. `/opt/OV/bin/ovstart -c`

Managing Overlapping IP Addresses in NAT Environments

NNMi helps you manage areas of your network that include Network Address Translation (NAT) domain implementations (potentially causing duplicate IP addresses, and requiring NNMi configuration for handling the NAT internal/external IP address pairs). NNMi administrators identify each NAT domain by creating a Tenant definition. NNMi identifies each Node by using a Tenant / IP address pair. Addresses are not considered duplicates unless they are duplicated within one Tenant's group of Nodes.

Note: Duplicate IP addresses outside of the context of NAT domain integrations: If your network includes firewall or load-balancer devices that have duplicate IP addresses / MAC addresses (such as virtual instances hosted on a physical device). The NNMi administrator populates a configuration file with the `sysObjectId` values of the firewall and load-balancer. Then, NNMi successfully acknowledges each instance of a Node object having those `sysObjectId` values (rather than merging all as if they were the same Node object).

What is NAT?

Network Address Translation (NAT) is typically used to interconnect a local network to the external (public) Internet. Specifically, NAT translates IP header information, substituting external (public) addresses for internal addresses in IP packets that need to transit the public network. NAT accomplishes this by providing either a static or dynamic external IP address. Network Address Translation is used as an Internet security measure, by never using the sender's IP address for Internet access.

Network Address Translation technology was developed as a solution for the ever-increasing need for more IPv4 addresses. Certain ranges of IP addresses (described in RFC 1918) are designated as internal only, in other words, not routable over the Internet. Anyone can use those addresses for private networks, reducing the number of public addresses that must be purchased.

What are the Benefits of NAT?

Some benefits of NAT include:

- Reuse of private IP addresses
- Enhancing security for private networks by keeping internal addressing private from the external network
- Connecting a large number of hosts to the global Internet using a smaller number of public (external) IP address, thereby conserving IP address space

What Types of NAT are Supported?

NNMi supports the following types of NAT protocols:

- **Static NAT**—A type of NAT in which an internal IP address is mapped to an external IP address, and the external address is always the same IP address (in other words, each Node has a static internal/external address pair). This permits an internal host, such as a Web server, to have a private IP address and still be reachable over the Internet.
- **Dynamic NAT**—A type of NAT in which mappings between external and internal addresses can change with each session. The internal IP address is dynamically mapped to a external IP address, drawing from a pool of available public IP addresses. Typically, the network's NAT gateway router keeps a table of registered public IP addresses, and when an internal IP address requests access to the Internet, the router chooses an IP address that is not currently being used by another internal IP address.
- **Dynamic Port Address Translation (PAT)**, also referred to as **Network Address and Port Translation (NAPT)** — A type of NAT that not only dynamically provides the external IP address but also dynamically provides the port number. Translating the address and the port number allows a single external address to be used for multiple simultaneous internal address conversations over the Internet.

How is NAT Implemented in NNMi?

NNMi manages NAT environments by identifying each Node using 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 each customer. Within that logical Tenant grouping:

- NNMi administrators use Discovery Seeds to identify the Tenant's member Nodes using a Tenant/IP address pair.
- 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) need to see.
- Security Groups determine how many Tenants an NNMi user can see. The assigned Security Group can include Nodes from more than one Tenant. For more information, see ["NNMi Security and Multi-Tenancy Configuration" on page 370](#).

Tip: A best practice is to have no duplicate Domain Name System (DNS) names across all NAT domains in your network management environment.

Depending on which NAT protocol you are using, the NNMi implementation method and requirements vary. For example, use of dynamic NAT or PAT would require additional hardware and licenses. See the appropriate sections based on your type of NAT protocol:

- "Static NAT Considerations" below
- "Dynamic NAT and PAT Considerations" on page 354

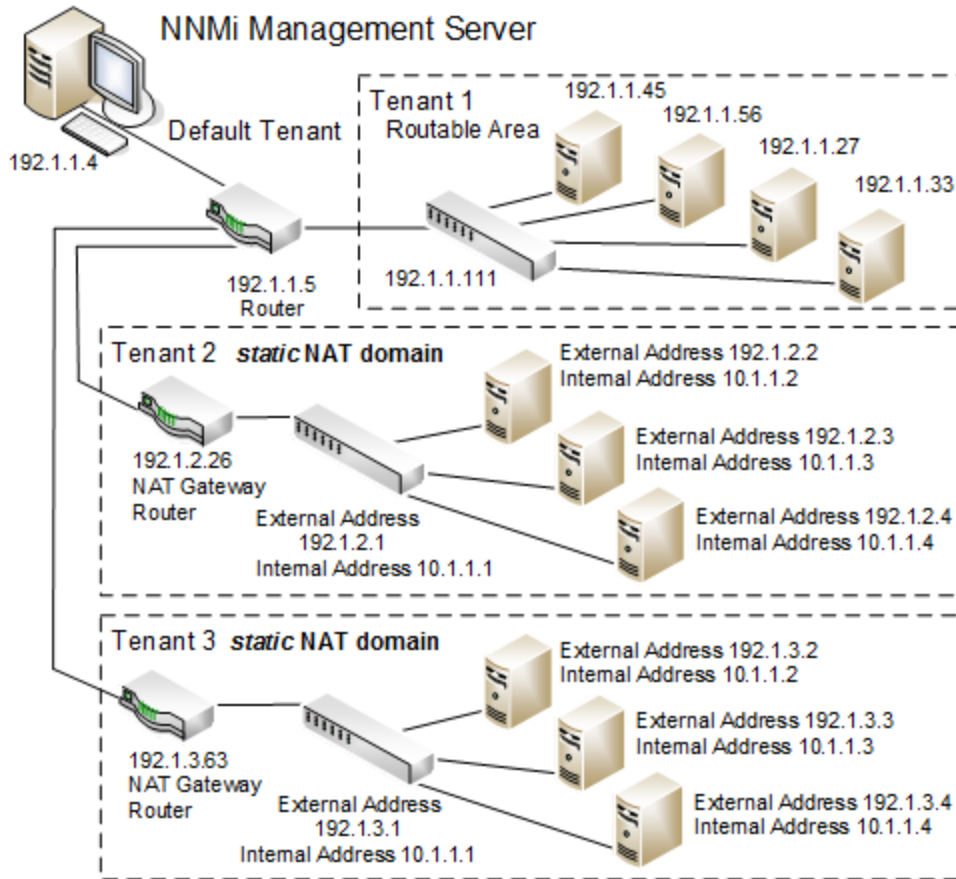
Then see "Deploy NNMi in a Network Address Translation (NAT) Environment" on page 357 for details.

Static NAT Considerations

Any number of static NAT instances can be monitored by one NNMi management server, as long as each instance is configured with a unique tenant. For more information on tenancy, see "NNMi Security and Multi-Tenancy" on page 360 and *Configure Tenants* in the NNMi help.

See the following diagram for an example of a static NAT configuration.

Example Static NAT Configurations



Note: Nodes that belong to the default tenant can have Layer 2 connections to any node in any tenant.

Nodes within any tenant other than the default tenant can have Layer 2 connections only to devices within the same tenant or the default tenant.

Subnets are tenant specific (in other words, subnets do not span tenants). The benefit here is that you can use the same subnet on different tenants.

Router Redundancy Groups (RRGs) cannot span tenants.

Tip: Assign any infrastructure device that interconnects 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) need to see.

Note: Devices within the default Security Group are visible from all views. To control access to a device, assign that device to a Security Group other than default Security Group.

Hardware and Software Requirements and Static NAT

There are no special hardware or software requirements for managing static NAT domains. One NNMi management server can manage any number of static NAT domains with either NNMi, NNMi Advanced, NNMi Premium, or NNMi Ultimate.

Overlapping IP Address Mapping

When the NNMi management server is outside of that static NAT domain, there are benefits to using Overlapping Address Mappings to identify each static NAT internal/external IP address pair. NNMi uses the mapping's External Address/Internal Address pairs in the following ways for static NAT domains:

- Node forms display a Mapped Address attribute value
- Communication and Monitoring processes are enhanced. This ensures that NNMi can successfully calculate state and status for each static NAT Node's SNMP Agent and managed IP addresses (see also "[NNMi Calculations for State and Status](#) " on page 359):
 - NNMi can accurately use the Monitoring Configuration Setting for ICMP Fault Monitoring's IP Address Fault Polling.
 - NNMi can determine accurate Layer 2 and Layer 3 connectivity for non-SNMP Nodes by using ICMP ping requests (in addition to SNMP queries).
- NNMi accurately determines SNMP Trap source Nodes when the traps originate from NAT domains. If SNMPv1 is used in your network, see also SNMP Traps in Static NAT Environments on page 240.
- Custom Incident Attributes are accurately calculated:
 - `cia.agentAddress` = The external IP address (public address).
 - `cia.internalAddress` = The internal IP address of the incident's Source Node.

Note: If you are configuring NNMi for areas of your network management domain that use dynamic NAT or PAT, do not use the Overlapping IP Address Mapping form. See "[Dynamic NAT and PAT](#)"

[Considerations" on page 354.](#)

Private IP Address Ranges

The Internet Engineering Task Force (IETF) and Internet Assigned Numbers Authority (IANA)'s reserved the following IP address ranges for private networks, for example enterprise local area networks (LANs), corporate offices, or residential networks.

IPv4 private address ranges (RFC 1918):

- 10.0.0.0 – 10.255.255.255 (24-bit block)
- 172.16.0.0 – 172.31.255.255 (20-bit block)
- 192.168.0.0 – 192.168.255.255 (16-bit block)

IPv6 private address ranges:

- fc00::/7 address block = RFC 4193 Unique Local Addresses (ULA)
- fec0::/10 address block = deprecated (RFC 3879)

Communication and Static NAT

NNMi successfully communicates through the static NAT firewall by automatically using any available Overlapping Address Mappings to determine the Tenant / External IP Address pair for static NAT communications. For information about the benefits, see ["Overlapping IP Address Mapping" on the previous page.](#)

Administering ICMP Polling of the Management Address 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, use the NAT address (the public IP address) 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 `ipAddressTable`.

NNMi provides this feature to avoid generating false node down incidents and a better root cause analysis.

Enabling ICMP Polling of the Management Address in a NAT Environment

By default, NNMi automatically enables ICMP management address polling for all nodes, including those nodes residing in a NAT environment. If you have a NAT environment, it is highly recommended that you do not disable this setting.

To enable ICMP management address polling (if it is disabled), do the following:

1. From the workspace navigation panel, select the **Configuration** workspace, expand the **Monitoring** folder, select **Monitoring Configuration**, and locate the **Default Settings** tab.
2. Enable ICMP Management Address Polling. See *Set Default Monitoring* in the NNMi help.

View the information NNMi displays after performing **Actions->Monitoring Settings** for SNMP Agents. The displayed information indicates whether NNMi has the management address polling enabled.

When ICMP Management Address Polling is enabled, NNMi changes as follows:

- The Agent ICMP State field appears in the following forms:
 - Node form
 - SNMP Agent form
 - SNMP Agent table views
- NNMi changes the display location of the management address ICMP state. NNMi also changes the way it determines the SNMP agent status.

The following table shows the Agent ICMP and IP Address state polling actions that NNMi takes for the ICMP Management Address Polling and ICMP Fault Polling settings.

ICMP Configurations and Resulting State Polling

ICMP Management Address Polling	ICMP 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

The following table shows changes to the SNMP Agent Status determined by APA for the SNMP agent and ICMP responses.

Determining SNMP Agent Status

SNMP Agent Response	Management Address ICMP Response	SNMP Agent Status
Responding	Responding	Normal
Responding	Not Responding	Minor
Not Responding	Responding	Critical
Not Responding	Not Responding	Critical

With ICMP polling of the management address enabled, APA now considers the management address ICMP response and the SNMP agent response when generating conclusions and generating incidents.

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 a Discovery Seed (Tenant / IP address pair) to identify each Node within the NAT domain. 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:

- External IP address (public address from the External/Internal IP address pair)
- Tenant name

See the NNMi help for more information.

Note: When adding Discovery seeds (using the `nmmloadseeds.ovpl` command or the NNMi console) in a static NAT environment, be sure to use the node's external (public) IP address. For more information, see the `nmmloadseeds.ovpl` reference page, or the Linux man page.

Tip: A best practice is to not have duplicate Domain Name System (DNS) names.

Monitoring Configuration for Static NAT

Depending on your network environment, the NNMi administrator can choose to use the ICMP Fault Monitoring settings (see also "[NNMi Calculations for State and Status](#)" on page 359):

- **Monitoring Configuration > Node Settings** tab to configure monitoring for a Node Group. In the ICMP Fault Monitoring section, make your choices (see the NNMi online Help for more information):
 - Management Address Polling (enabled by default and highly recommended)
 - IP Address Fault Polling (optional)
- **Monitoring Configuration > Default Settings** tab. In the ICMP Fault Monitoring section, make your choices (see the NNMi online Help for more information):

Note: If your network environment also includes any dynamic NAT domains, Default settings might not be appropriate because you might want different settings for static NAT domains from those for dynamic NAT domains.

Traps and Static NAT

You must make changes to the managed nodes for the NNMi management server to receive SNMP traps from nodes behind the NAT gateway. This section covers two types of SNMP traps: SNMPv2c and SNMPv1.

Note that NNMi must unambiguously resolve the source address of each trap that it receives.

SNMPv2c Traps

The following table shows the format of an SNMPv2c trap, with the IP header forming the top section of the table and the SNMP Trap Protocol Data Unit (PDU) forming the lower section of the table.

SNMPv2c Trap Format

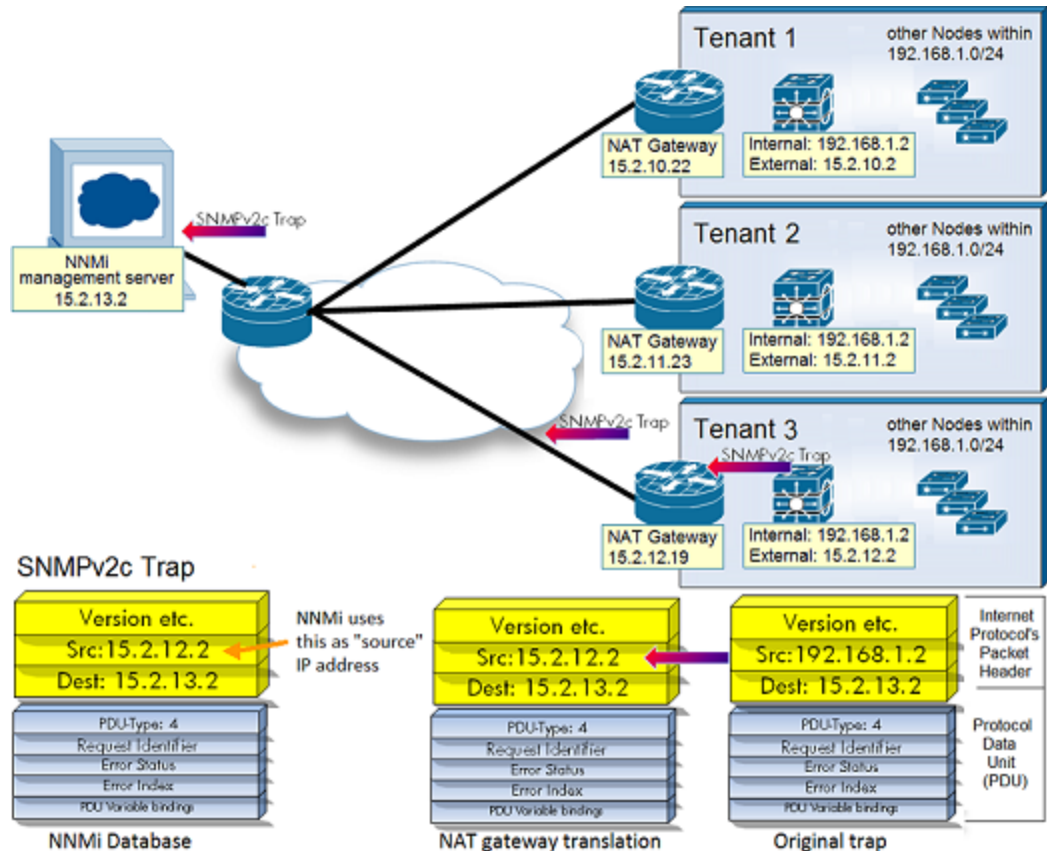
Version and other information
Source Address
Destination Address
PDU-Type: 4
Request Identifier
Error Status
Error Index
PDU Variable Bindings

SNMPv2c traps do not have an Agent Address field in the PDU; therefore, the only source field of the trap is within the IP packet header. NAT routers properly translate the source field.

On the source node, ensure that the interface associated with the private inside IP address sources all traps from devices behind the NAT router. Then, the NAT gateway can translate the trap to the correct public address.

The following diagram shows an example of correct translation from the NAT gateway. The NAT gateway properly translates a trap that begins with the source address of 192.168.1.2 to address 15.2.13.2. Then the NNMi management server correctly resolves this address.

SNMPv2c Example



SNMPv1 Traps

SNMPv1 traps embed the Agent Address inside the SNMP trap PDU. The following table shows the format of an SNMPv1 trap, with the IP header forming the top section and the SNMP trap PDU forming the lower section.

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 is embedded in the PDU rather than the header, usually the NAT router will not translate this value. You can enable NNMi to note the address in the header and ignore the Agent Address in the payload by doing the following:

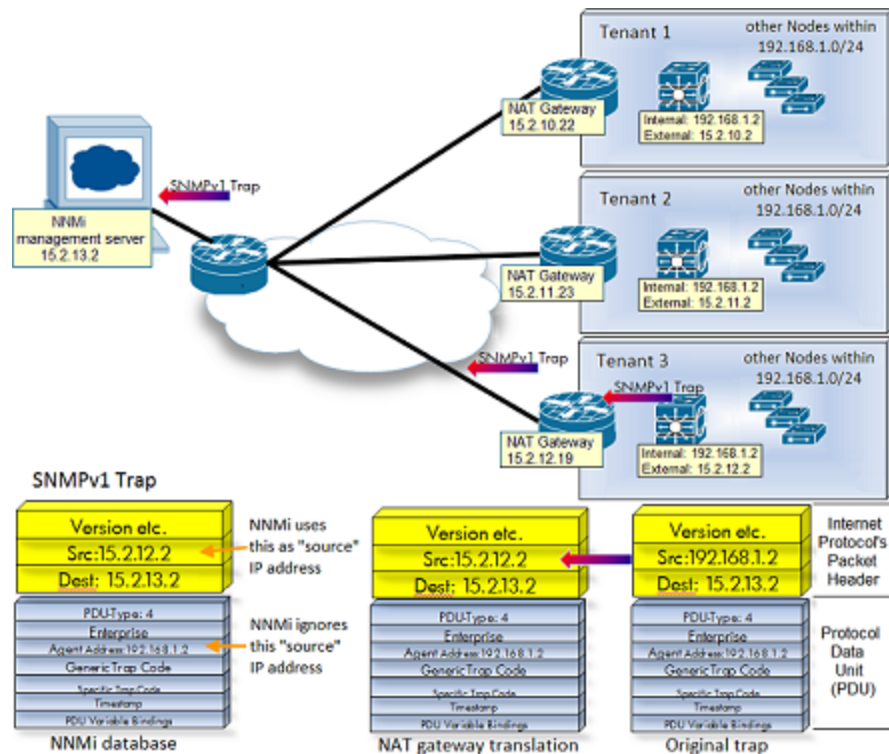
1. Edit the following file:
 - *Windows:* %NNM_PROPS%\nms-jboss.properties
 - *UNIX:* \$NNM_PROPS/nms-jboss.properties
2. Find the following line


```
#!/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 file; then restart NNMi.

The following diagram shows an example of an SNMPv1 trap where NNMi ignores the conflicting IP address fields.

SNMPv1 Example



Note: NNMi provides the following related Custom Incident Attributes (CIAs):

- `cia.agentAddress`—the IP address stored in the SNMPv1 trap data for the SNMP Agent that generated the trap.
- `cia.internalAddress`—If static NAT is part of your network management domain, the NNMi administrator can configure this attribute to show the internal IP address that is mapped to the external management address of the selected incident's Source Node.

The external management IP address (public address) must be mapped to this internal address (private address) using the Overlapping IP Address Mapping form. For more information, see the NNMi help.

Subnets and Static NAT

Note the following with regard to subnets and NAT:

- Subnets are tenant specific (in other words, subnets do not span tenants). The benefit here is that you can use the same subnet on different tenants.
- Subnet filters use a tenant and address pair.
- If you configure a subnet connection rule, the rule applies to all tenants. The members of the subnets must be unique across all tenants (each node assigned to only one tenant). A subnet connection rule can establish a link between the default tenant and another tenant. However, links between two tenants are not allowed unless one of them is the default tenant.

Global Network Management: Optional for Static NAT

The NNMi Global Network Management feature is *optional* when managing static NAT domains. Only one NNMi management server is required to manage any number of static NAT domains.

If using Global Managers and Regional Managers, 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.

Dynamic NAT and PAT Considerations

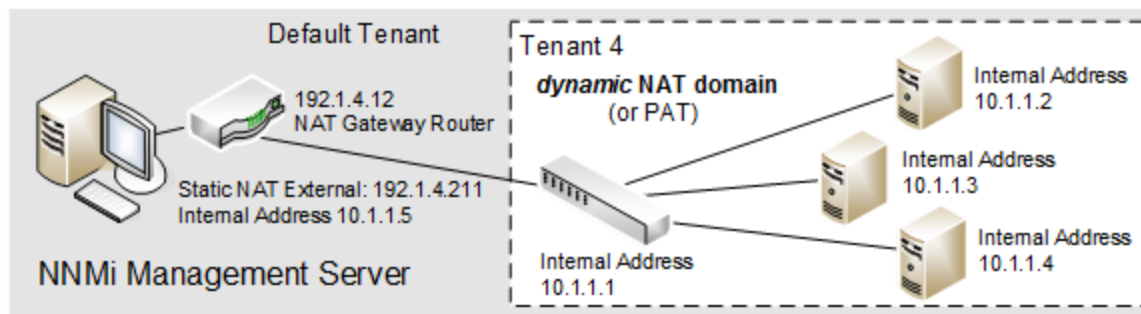
Each dynamic NAT or PAT domain requires its own NNMi management server. The NNMi management server must participate in a Global Network Management environment as a Regional Manager.

The NNMi administrator creates a Tenant definition to identify each NAT domain. Tenants must be unique within the entire NNMi Global Network Management configuration.

See the following two examples of a dynamic NAT configuration.

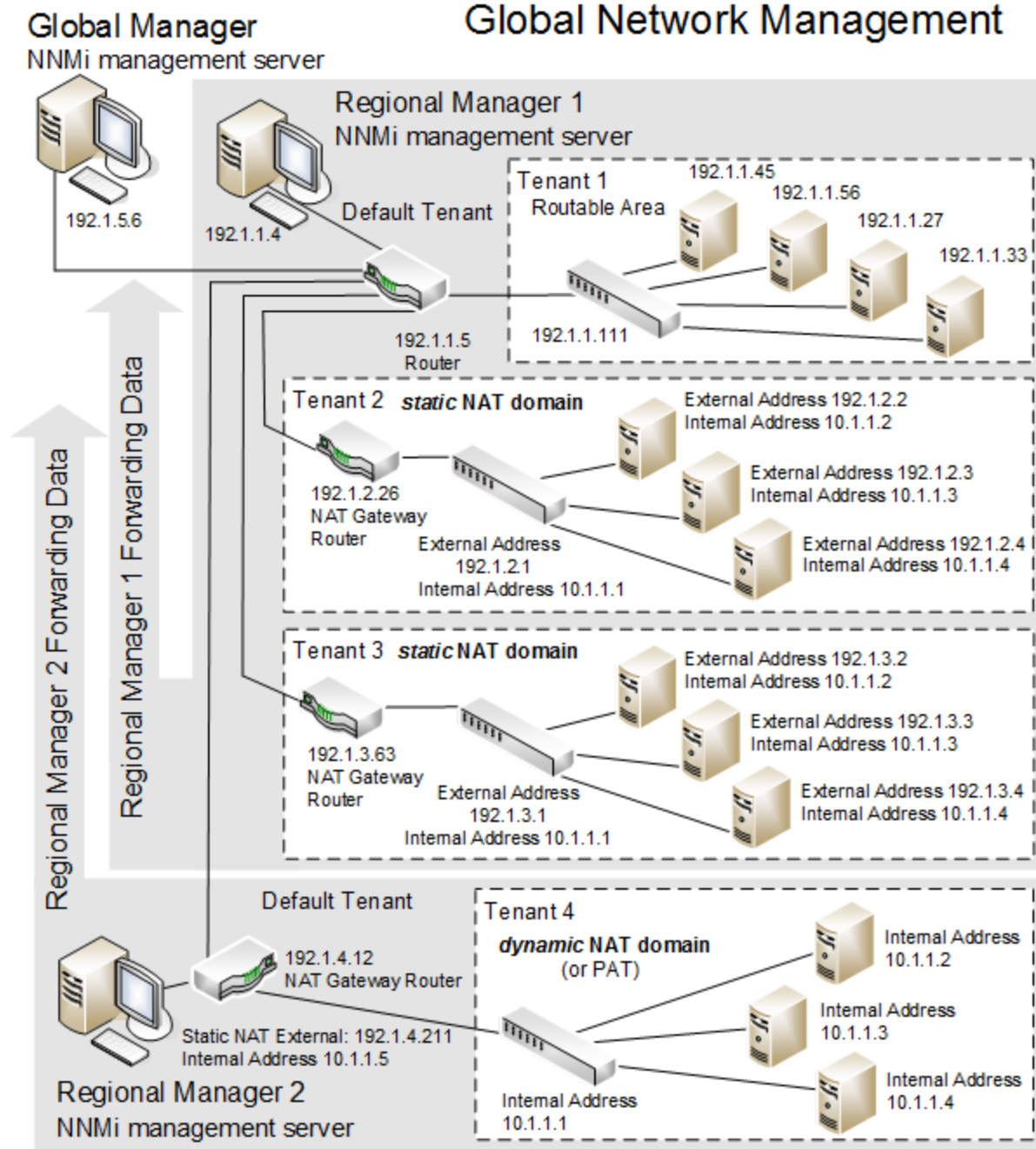
Note: If a Regional Manager is located behind a NAT firewall, its external (public) address must be static.

Example Dynamic NAT Configurations



See the following figure for an example of a Global Network Management configuration within a NAT environment.

Example Global Network Management Configuration within a NAT Environment



Devices that belong to the default tenant can have Layer 2 connections to any device in any tenant. Devices within any tenant other than default tenant can have Layer 2 connections only to devices within the same tenant or the default tenant.

Tip: Assign any infrastructure device that interconnects multiple NAT domains (such as the NAT gateway) to the default tenant. This ensures that NNMi displays the Layer 2 connections your workgroup (and customers) need to see.

Devices within the default Security Group are visible from all views. To control access to a device, assign that device to a Security Group other than default Security Group.

Hardware and Software Requirements and Dynamic NAT and PAT

NNMi Advanced, NNMi Premium, or NNMi Ultimate software is required for dynamic NAT and PAT environments.

An NNMi Regional Manager is required for each address domain configured with dynamic NAT or PAT.

Discovery Configuration for Dynamic NAT and PAT

The NNMi administrator must create a Tenant definition to identify each dynamic NAT domain within your network management environment. Those Tenant names must be unique within the entire NNMi Global Network Management configuration.

Spiral Discovery requires a Discovery Seed (Tenant / IP address pair) to identify each Node within the NAT domain. The NNMi administrator must create a Discovery Seed for each Node in the dynamic NAT domain. A Discovery Seed must provide the following information for each Node:

- Internal IP address (public address from the External Address/Internal Address pair)
- Tenant name

Note: When adding Discovery seeds (using the `nmloadseeds.ovpl` command or the graphical user interface) in a dynamic NAT or PAT environment, be sure to use the node's internal IP address.

For more information, see the *nmloadseeds.ovpl* reference page, the Linux man page, or the NNMi help.

Monitoring Configuration for Dynamic NAT

Depending on your network environment, the NNMi administrator can choose to use the ICMP Fault Monitoring settings (see also "[NNMi Calculations for State and Status](#)" on page 359):

- **Monitoring Configuration > Node Settings** tab to configure monitoring for a Node Group. In the ICMP Fault Monitoring section, make your choices (see the NNMi online Help for more information):
 - Management Address Polling (enabled by default and highly recommended)
 - IP Address Fault Polling (optional)
- **Monitoring Configuration > Default Settings** tab. In the ICMP Fault Monitoring section, make your choices (see the NNMi online Help for more information):

Note: 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.

Subnets and Dynamic NAT and PAT

When using subnets in a Dynamic NAT or PAT environment, note the following:

- Subnets are tenant specific (in other words, subnets do not span tenants).

Tip: You can use the same subnet on different tenants.

- Subnet filters use a tenant/address pair.
- If you configure a subnet connection rule, the rule applies to all tenants. The members of the subnets must be unique across all tenants (each node assigned to only one tenant). A subnet connection rule can establish a link between the default tenant and another tenant. However, links between two tenants are not allowed unless one of them is the default tenant.

Global Network Management: Required for Dynamic NAT and PAT

The NNMi Global Network Management feature is required when managing dynamic NAT domains. Each dynamic NAT or PAT domain needs its own NNMi Regional Manager.

At least one static or routable (non-translated) address must exist per NNMi Regional Manager. This enables NNMi management servers to communicate with each other, keeping communications internal and secure.

If a regional manager is behind a NAT firewall, its external address must be static.

Deploy NNMi in a Network Address Translation (NAT) Environment

Follow these steps to deploy NNMi in a NAT environment:

1. Identify and make a list of each NAT domain in your network management environment.
2. Determine which type of supported NAT 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 special considerations:

["Static NAT Considerations" on page 346](#)

["Dynamic NAT and PAT Considerations" on page 354](#)

4. Use the NNMi **Configuration > Discovery > Tenants** workspace to define a unique Tenant name for each NAT domain.

Note: If using Global Network Management in your deployment, this name must be unique across all NNMi management servers (Regional Managers and the Global Manager).

5. Decide which 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 ["Overlapping IP Address Mapping" on page 347](#).

Provide the following information:

- Tenant name
- External IP address
- Internal IP address

Use either the NNMi **Configuration > Discovery > Overlapping Address Mappings** workspace or the `nnmloadipmappings.ovp1` command line tool.

See the NNMi online Help for details.

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 Address. Therefore, for **Configuration > Communication Configuration** settings, use the appropriate Preferred Management Address setting (NAT's External or Internal IP address).
8. Verify Monitoring Configuration settings for NAT in your network environment:
 - ["Monitoring Configuration for Static NAT" on page 350](#)
 - ["Monitoring Configuration for Dynamic NAT" on page 356](#)

See the NNMi online Help if you need more information about Monitoring Configuration.
9. Configure a Discovery Seed for each Node.

Note: Assign any infrastructure device that interconnects multiple NAT domains (such as the NAT gateway router) to the Default Tenant.

Use either the NNMi **Configuration > Discovery > Seeds** workspace or the `loadseeds.ovp1` command line tool:

- If the NNMi management server is inside the internal IP address space, configure Discovery Seeds using the Internal IP address:
 - Hostname/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:
 - Hostname/IP (use the External IP address)
 - Tenant name

See the NNMi online Help for details.

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 control which team members / customers can see each Node in the NNMi console. Use NNMi's **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 NNMi maps as expected. If not:
 - Verify that both Nodes involved in the connection have proper Tenant assignments (Default Tenant or other tenant).
 - Verify that your **Configuration > Discovery Configuration's Subnet Connection Rules** tab settings are correct.

- To force NNMi to add connections that are not automatically found, use the `nnmconnect.ovpl` command line tool. See the NNMi online **Help > NNMi Documentation Library > Reference Pages** for details.
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. For static NAT domains only: Configure the SNMP Agent on each static NAT Node to ensure that the interface associated with the NNMi Overlapping Address Mappings Internal Address 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 "[Traps and Static NAT](#)" on page 350.

NNMi Calculations for State and Status

By default, NNMi automatically enables ICMP polling of each Node's management address, including those 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, it is highly recommended that you do not disable this setting.

Note: In the **Inventory > SNMP Agent** view, select an SNMP Agent and use the **Actions > 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 ICMP Fault Monitoring settings. The first row in the table shows the NNMi default settings.

Monitoring Configuration Settings and the Resulting State Poller Behavior

ICMP Fault Monitoring Settings		Resulting NNMi Behavior	
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

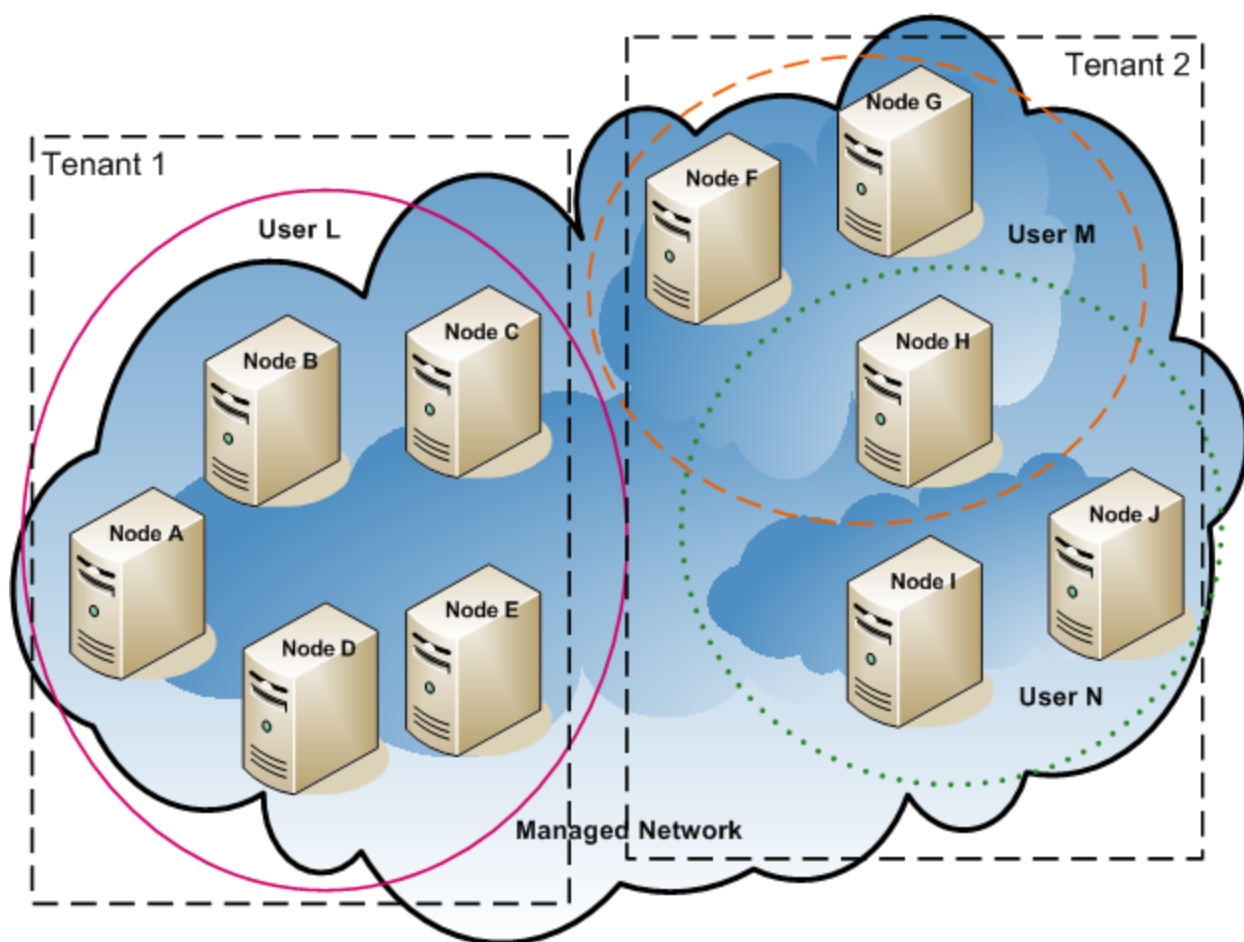
When **Management Address Polling** is enabled, NNMi considers both the management address's ICMP response and the SNMP Agent's response when calculating conclusions and generating incidents.

The following table shows the SNMP Agent Status calculations determined by the combined ICMP and SNMP responses.

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

NNMi Security and Multi-Tenancy



Note: NNMi uses tenancy to support networks with overlapping address domains that may exist within static Network Address Translation (NAT), dynamic NAT, or dynamic Port Address Translation (PAT) areas of your network management domain. If you have such networks, put the overlapping address domains into different tenants (this is done using seeded discovery). See ["Managing Overlapping IP Addresses in NAT Environments"](#) on page 344 and the NNMi help for more information.

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.

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. It contains the following topics:

- ["Effects of Limiting Object Access" below](#)
- ["The NNMi Security Model" on the next page](#)
- ["The NNMi Tenant Model" on page 367](#)
- ["NNMi Security and Multi-Tenancy Configuration" on page 370](#)
- ["NNMi Security, Multi-Tenancy, and Global Network Management \(GNM\)" on page 378](#)
- ["Including Select Interfaces in NPS Reports" on page 381](#)

See also the *HPE Network Node Manager i Software Step-by-Step Guide to Using Security Groups White Paper*.

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 their 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.
 - A NNMi console user sees only those node groups for which they can access at least one node in the group.
 - For Network Performance Server (NPS) reports, the NNMi administrator can selectively override access control inheritance on interfaces. For more information, see ["Including Select Interfaces in NPS Reports" on page 381](#).
- 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.
 - For the NNM iSPI for MPLS and the NNM iSPI for IP Multicast, when maps and path views include nodes to which the NNMi console user does not have access, the NNM iSPI displays only the connecting interface and the name of the node. The icons for the inaccessible nodes are white to indicate that status and detailed information are not available for these nodes.

- For the NNM iSPI for IP Telephony, when maps and path views include nodes to which the NNMi console user does not have access, the NNM iSPI displays only the connecting interface and the name of the node. The icons for the inaccessible nodes show the NNMi status, but all attempted actions fail.
- 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).

Note: 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 they have 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 and NNM iSPI table views and forms, 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 they have access.
 - An NNMi console user can view MIB data for nodes to which they know the SNMP community string.
- NNMi console URLs:

Users must log on 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.

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 and Network Performance Server reports 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.

Security Groups

In the NNMi security model, user access to nodes is controlled indirectly through 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 Level 2 Operators
 - NNMi Level 1 Operators
 - NNMi Guest Users

This mapping is required for NNMi console access and determines which actions 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.

Note: 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.

Note: The NNMi Global Operators User Group (`g1oba1ops`) grants access to topology objects only. A user must be assigned to one of the other User Groups (`leve12`, `leve11`, or `guest`) to access the NNMi console.

The administrator should not map the `g1oba1ops` User Group to any security group because this User Group is, by default, mapped to all security groups.

- Zero or more 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 Default Security Group is the initial security group assignment for all nodes. By default, all users can see all objects in the Default Security Group. The NNMi administrator can configure which nodes are associated with the Default Security Group and which users 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 which users can access the incidents associated with the Unresolved Incidents security group.

All sensors inherit the security group assignment of the node.

Note: The following best practices apply to NNMi security configuration:

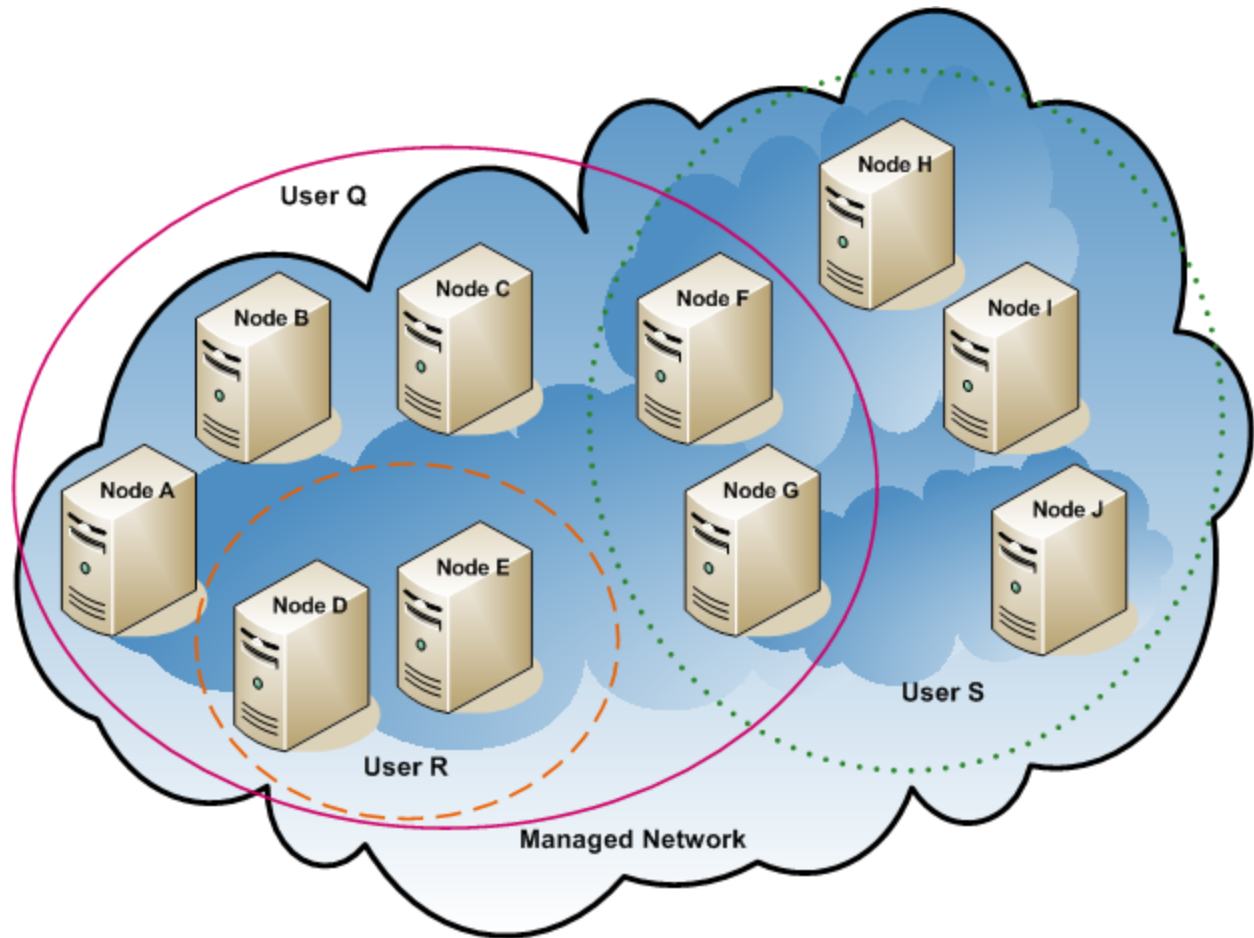
- Map each user account to only one preconfigured NNMi user group.
- Do not map the preconfigured NNMi user groups to security groups.
- Because any user account mapped to the NNMi Administrators user group receives administrator-level access to all objects in the NNMi database, do not map this user 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.

Example Security Group Structure

The three ovals in the following diagram indicate the primary groupings for which users need to view the nodes in this example NNMi topology. 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.

[Example Security Group Mappings](#) 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.) [Example User Account Mappings](#) lists the mappings for several user accounts and the user groups for this topology.

Example Topology for User Access Requirements



Example Security Group Mappings

Security Group	Nodes of Security Group	User Group	Object Access Privilege
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

Example Security Group Mappings, continued

Security Group	Nodes of Security Group	User Group	Object Access Privilege
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

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 nodes in the pink oval (solid line).
	UG1 Level 2	A, B, C	
	UG2 Level 2	D, E	
	UG3 Level 2	F, G	
User R	NNMi Level 1 Operators	none	This user has operator level 1 access to the nodes in the orange oval (dashed line).
	UG2 Level 1	D, E	
User S	NNMi Level 2 Operators	none	This user has operator level 2 access to the nodes in the green oval (dotted line).
	UG3 Level 2	F, G	
	UG4 Level 2	H, I, J	

Example User Account Mappings, continued

User Account	User Groups	Node Access	Notes
User T	NNMi Level 2 Operators	none	This user has access (with varying privilege levels) to all nodes in the example topology.
	UG1 Guest	A, B, C	
	UG2 Administrator	D, E	This user has administrative access to nodes D and E but cannot see the menu items for tools that require administrative access. If this user has access to the NNMi management server, this user can run command-line tools that require administrative access against nodes D and E only.
	UG3 Level 2	F, G	
	UG4 Level 1	H, I, J	

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 and Network Performance Server reports 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.
- Provides for management of overlapping address domains when address translation protocols are used.

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.

Note: Any number of static Network Address Translation (NAT) instances can be monitored by one NNMi management server, as long as each instance is configured with a unique tenant. See ["Managing Overlapping IP Addresses in NAT Environments"](#) on page 344, and the NNMi help, for more information.

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 all other discovered nodes (those included in an auto-discovery rule but not seeded directly) to the Default Tenant. 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.

Tip: Changing the tenant assignment of a node does not automatically change the security group assignment.

NNMi provides the Default Tenant. By default, 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.

Note: The following best practices apply to NNMi tenant configuration:

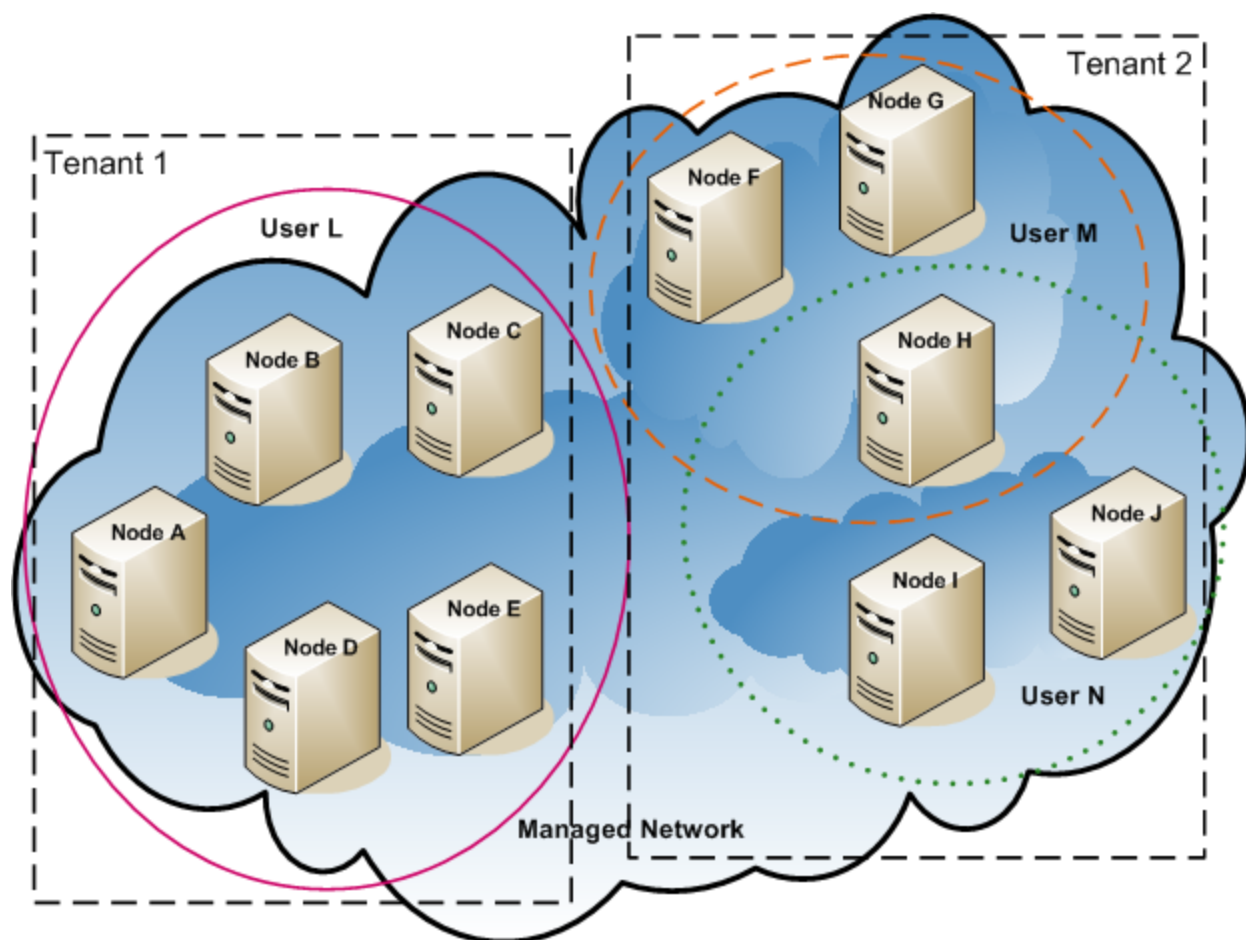
- For a small organization, a single security group per tenant is probably sufficient.
- You might want to subdivide a large organization into multiple security groups.
- To prevent users from accessing nodes across organizations, ensure that each security group includes nodes for only one tenant.

Example Tenant Structure

The following diagram shows an example NNMi topology containing two tenants, represented by the rectangles. The three ovals 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.

[Example Security Group Mappings for Multiple Tenants](#) 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.) [Example User Account Mappings for Multiple Tenants](#) lists the mappings for several user accounts and the user groups for this topology.

Example Topology for Multiple Tenants



Example Security Group Mappings for Multiple Tenants

Security Group	Nodes of Security Group	User Group	Object Access Privilege
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 Level 2	Object Operator Level 2
		T2_a Level 1	Object Operator Level 1
		T2_a Guest	Object Guest

Example Security Group Mappings for Multiple Tenants, continued

Security Group	Nodes of Security Group	User Group	Object Access Privilege
T2 SGb	H	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
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

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 nodes in the pink oval (solid line), which groups all nodes in Tenant 1.
	T1 Level 2	A, B, C, D, E	
User M	NNMi Level 1 Operators	none	This user has operator level 1 access to the nodes in the orange oval (dashed line), which groups a subset of the nodes in Tenant 2.
	T2_a Level 1	F, G	
	T2_b Level 1	H	
User N	NNMi Level 2 Operators	none	This user has operator level 2 access to the nodes in the green oval (dotted line), which groups a subset of the nodes in Tenant 2.
	T2_b Level 2	H	
	T2_c Level 2	I, J	

NNMi Security and Multi-Tenancy Configuration

Note: Any number of static Network Address Translation (NAT) instances can be monitored by one NNMi management server, as long as each instance is configured with a unique tenant. See ["Managing Overlapping IP Addresses in NAT Environments"](#) on page 344, and the NNMi help, for more information.

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 Group** 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.

Tip: 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

The 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. It is recommended to grant each user account the NNMi role that matches the highest object access privilege for that user's topology objects.

Note: 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. (Level 1 Operators have less access privileges than Level 2 Operators.) 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 more information, see ["Integrating NNMi with a Directory Service through LDAP" on page 308](#).

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** page presents a list of unsaved changes from the current wizard session. It also identifies potential problems with the security configuration.

Note: The **Security Wizard** is for NNMi security configuration only. It does not include tenant information.

For information about using the **Security Wizard**, click the NNMi help links within the wizard.

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 information about using these forms, see the 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 `nmmsecurity.ovpl` command.

The following NNMi console views are available under **Security** in the **Configuration** workspace. These views contain NNMi security configuration information:

- **User Accounts**

- Each **User Account** 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 Group** form describes one NNMi user group and shows the user accounts and security groups mapped to the user group. The mapping information is read-only.

- **User Account Mappings**

- 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 log on 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 Group** 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 Mappings**

- 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 recreate it.

Command line

The `nmsecurity.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 `nmsecurity.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 `nmsecurity.ovpl` command. The command can also accept UUIDs generated outside of NNMI.

Tip: Because security group and tenant names do not need to be unique, specify the security group or tenant UUID as input to the `nmsecurity.ovpl` command.

The following example script uses the `nmsecurity.ovpl` command to create the security configuration for two user accounts and five nodes.

```
#!/bin/sh

# create two users

nmsecurity.ovpl -createUserAccount -u user1 -p password -role level1
nmsecurity.ovpl -createUserAccount -u user2 -p password -role level2

# create two user groups

nmsecurity.ovpl -createUserGroup local1
nmsecurity.ovpl -createUserGroup local2

# assign the user accounts to the new user groups

nmsecurity.ovpl -assignUserToGroup -user user1 -userGroup local1
nmsecurity.ovpl -assignUserToGroup -user user2 -userGroup local2

# create two security groups

nmsecurity.ovpl -createSecurityGroup secgroup1
nmsecurity.ovpl -createSecurityGroup secgroup2

# assign the new user groups to the new security groups

nmsecurity.ovpl -assignUserGroupToSecurityGroup -userGroup local1
    -securityGroup secgroup1 -role level1

nmsecurity.ovpl -assignUserGroupToSecurityGroup -userGroup local2
    -securityGroup secgroup2 -role level2

# assign nodes to security groups

nmsecurity.ovpl -assignNodeToSecurityGroup -node mplspe01 -securityGroup secgroup1
nmsecurity.ovpl -assignNodeToSecurityGroup -node vwan_router-1 -securityGroup secgroup1
nmsecurity.ovpl -assignNodeToSecurityGroup -node vwan_router-2 -securityGroup secgroup1
nmsecurity.ovpl -assignNodeToSecurityGroup -node data_center_1 -securityGroup secgroup2
nmsecurity.ovpl -assignNodeToSecurityGroup -node mplspe03 -securityGroup secgroup2
```

Configuring Tenants

Note: Any number of static Network Address Translation (NAT) instances can be monitored by one NNMi management server, as long as each instance is configured with a unique tenant. See ["Managing Overlapping IP Addresses in NAT Environments" on page 344](#), and the NNMi help, for more information.

NNMi provides the following ways to configure multi-tenancy:

- The **Tenant** form in the NNMi console is useful for working with individual tenants.
- The `nmmsecurity.ovpl` command-line interface 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. This high-level procedure describes one approach to configuring NNMi multi-tenancy.

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.
- 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:

1. Analyze your customer requirements to determine how many tenants are required in the NNMi environment.

It is recommended 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 the Unresolved Incidents security group.

Doing this step assures that users do not inadvertently obtain access to nodes they should not 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.

Tip: 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.

See "[Verifying the Configuration](#)" on the next page.

Configuring Security Groups

Tip: 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 is useful for visualizing the security configuration. The **View Summary of Changes** page presents a list of unsaved changes from the current wizard session. It also identifies potential problems with the security configuration.
- The forms in the NNMi console for individual security objects are useful for concentrating on one aspect of the security configuration at a time.
- The `nmsecurity.ovp1` command-line interface 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. This high-level procedure describes one approach to configuring NNMi security.

Tip: 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 the 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 the Unresolved Incidents security group.
Doing this step assures that users do not inadvertently obtain access to nodes they should not 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.
 - o If you are storing user group membership in the NNMi database, no users are mapped to these user groups yet.
 - o 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:
 - o Create a user account object for each user who can access the NNMi console. (The process of configuring user accounts depends on whether you are using a directory service for NNMi console logon.)
 - o Map each user account to one of the predefined NNMi user groups (for access to the NNMi console).
 - o 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 "[Verifying the Configuration](#)" below.
 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.

Verifying the Configuration

To verify that the security configuration is correct, verify each aspect of the configuration separately. This section describes some approaches to verifying the configuration. Other approaches are possible.

Note: NNMi provides reports of possible security configuration errors. Access these reports with **Tools > Security Reports** in the NNMi console and with the `-displayConfigReport` option to the `nmsecurity.ovpl` command.

Verify security group-to-node assignments

One approach to verifying that each node is assigned to the correct security group is to sort the **Nodes** or **Nodes (All Attributes)** inventory view by security group, and then examine the groupings.

Another approach is to use the `-listNodesInSecurityGroup` option to the `nmsecurity.ovpl` command.

Verify user group-to-security group assignments

One approach to verifying which user groups are mapped to each security group is to sort the **Security Group Mappings** view by user group or security group, and then examine the groupings. Also verify the object access privilege for each mapping.

Alternatively, on the **Map User Groups and Security Groups** page of the **Security Wizard**, select one user group or security group at a time to see the current mappings for that object.

Another approach is to use the `-listUserGroupsForSecurityGroup` option to the `nmsecurity.ovpl` command.

Verify 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 (listed from highest to lowest):

- NNMi Administrators
- NNMi Level 2 Operators
- NNMi Level 1 Operators
- NNMi Guest Users

All other user group assignments provide access to objects in the NNMi database.

Note: The NNMi Global Operators Users Group provides access to topology objects only. Unless a `globalops` user is also associated with a User Group with NNMi Console access (such as `level2`, `level1`, or `guest`), that user will not be able to access the NNMi console.

Users without NNMi console access are listed on the **View Summary of Changes** page of the **Security Wizard**. The **Tools > Security Reports** menu item and the `-displayConfigReport usersWithoutRoles` option to the `nmsecurity.ovpl` command also provide this information.

Note: Each **Tools** and **Action** menu item provided in the NNMi Console is associated with a default NNMi role. (To determine the default NNMi Role assigned to each Action menu item, see *Actions Provided by NNMi* in the NNMi help.) If you change the setting for a menu item provided by NNMi to a role that is a lower level role than the default NNMi role assigned to the menu item, NNMi ignores that change. Any User Group with the lower level role than the default NNMi role cannot access the menu item.

Verify user-to-user group assignments

One approach to verifying user group membership is to sort the **User Account Mappings** view by user account or user group, and then examine the groupings.

Alternatively, on the **Map User Accounts and User Groups** page of the **Security Wizard**, select one user account or user group at a time to see the current mappings for that object.

Another approach is to use the `-listUserGroups` and `-listUserGroupMembers` options to the `nmsecurity.ovpl` command.

Verify 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.

Verify current user settings

To verify the NNMi console access for the currently logged-on user, click **Help > 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 (NNMi Administrators, NNMi Level 2 Operators, NNMi Level 1 Operators, and NNMi Guest Users) to which the user is mapped. This mapping determines which actions are available within the NNMi console.
- User groups mapped to this user name. This list includes predefined NNMi user group that sets the NNMi role and any other user groups that provide access to objects in the NNMi database.

Exporting the NNMi Security and Multi-Tenancy Configuration

The following table describes the configuration areas (available with `nnmconfigexport.ovpl -c`) 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.

NNMiSecurity and Multi-Tenancy Configuration Export Areas

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.

NNMi Security, Multi-Tenancy, and Global Network Management (GNM)

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.

Note: Each group of dynamic Network Address Translation (NAT) or dynamic Port Address Translation (PAT) requires an NNMi regional manager, in addition to a tenant that is unique within the entire NNMi global network management configuration. See ["Managing Overlapping IP Addresses in NAT Environments" on page 344](#). See also the NNMi help.

Tip: 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 group with the same UUID as on another NNMi management server. Following this recommendation ensures that each tenant and security group has the same UUID within the GNM environment.

Note: This best practice becomes a *required* part of the configuration if users will be launching NPS reports from the global manager.

Note: 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.

Tip: 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`

Initial GNM Configuration

After 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).

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.

Tip: Suggestions for simplifying security configuration on the global manager include:

- 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.

GNM Maintenance

The following table describes how changes to a node's tenant or security group assignment on a regional manager affect the global manager.

Global Manager Effects of Configuration Changes on a Regional Manager

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.

Global Manager Effects of Configuration Changes on a Regional Manager, continued

Action	Effect
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.

Including Select Interfaces in NPS Reports

The Network Performance Server (NPS) is the database server installed with the NNMI iSPI Performance for Metrics software.

By default, all components of a node are in the same security group as the node. For individual interfaces, you can override this default behavior and assign an interface to a different security group. The purpose of this override is to generate tenant-specific reports that include the appropriate interfaces for that tenant (customer) on shared devices. In this way, each customer can see the interface information for their interfaces but cannot see the other interfaces on the device.

Note: The security group override only affects NPS reports. It has no impact on what users can see and do in the NNMI console.

To change the security group assignment for an interface, on the **Custom Attributes** tab of an **Interface** form or with the `nnmloadattributes.ovpl` command, add the `InterfaceSecurityGroupOverride` custom attribute to that interface. Set the value of this custom attribute to the UUID of the security group. For example:

```
InterfaceSecurityGroupOverride=0826c95c-5ec8-4b8c-8998-301e0cf3c1c2
```

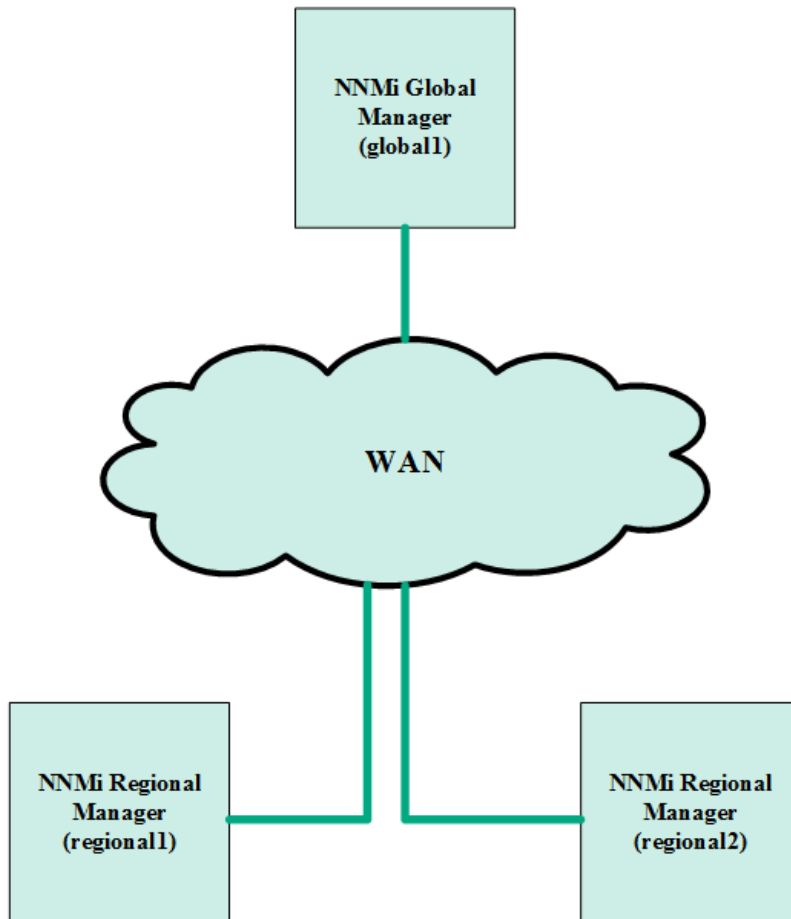
Note: An interface can belong to only one security group at a time. Setting the `InterfaceSecurityGroupOverride` custom attribute on an interface breaks the association between that interface and the security group to which its node belongs.

Configuring Single Sign-On for Global Network Management

You can configure NNMI single sign-on (SSO) to facilitate access to NNMI regional managers from an NNMI global manager.

Note: You must configure single sign-on before connecting regional managers from a global manager. See ["Using Single Sign-On \(SSO\) with NNMI" on page 280](#) for more information.

Global Network Management



The SSO feature communicates a user name among NNMI management servers, but not passwords or roles. For example, NNMI associates the same username on one NNMI management server (`global1`) with a different role on other NNMI management servers (`regional1` or `regional2`). Any of these three NNMI management servers could associate a different password with the same username.

If a global and regional manager resides in the same management domain, and you do not copy the *Initialization String* value from the global NNMI management server to the regional NNMI management server as shown in [step 4](#), you could have NNMI console access problems. To avoid this, either configure SSO correctly using the following steps, or disable SSO as described in ["Disabling SSO" on page 284](#).

To configure SSO to work with the global network management feature, complete the following steps:

Note: Global and regional managers need to be in the same domain.

1. Open the following file on `global1`, `regional1`, and `regional2`:
 - *Windows:* `%NNM_PROPS%\nms-ui.properties`
 - *Linux:* `$NNM_PROPS/nms-ui.properties`

- On `global1`, `regional1`, and `regional2`, look for a section in the file that resembles the following:

```
com.hp.nms.ui.sso.isEnabled = false
```

Change this as follows:

```
com.hp.nms.ui.sso.isEnabled = true
```

- Locate the SSO NNMi initialization string for `global1`. Look for a section in the `nms-ui.properties` file that resembles the following:

```
com.hp.nms.ui.sso.initString=Initialization String
```

- Copy the value of *Initialization String* from the `nms-ui.properties` file on `global1` to the `nms-ui.properties` files on `regional1` and `regional2`. All of the servers must use the same value for *Initialization String*. Save your changes.

Note: NNMi supports copying the *Initialization String* value from the global NNMi management server to the regional NNMi management servers. In this step, you copied the *Initialization String* value from the global manager to the two regional managers. Always copy the *Initialization String* value from the global manager to the regional managers if you want to use SSO with the global network management feature.

Note: If a global and regional manager resides in the same management domain, and you do not copy the *Initialization String* value from the global NNMi management server to the regional NNMi management server, disable SSO to avoid NNMi console access problems. See "[Disabling SSO](#)" on page 284 for more information.

- If `global1`, `regional1`, and `regional2` are in different domains, modify the `protectedDomains` content. To do this, look in the `nms-ui.properties` file for a section that resembles the following:

```
com.hp.nms.ui.sso.protectedDomains=group1.mycompany.com
```

Suppose `global1` is in `global1.company1.com`, `regional1` is in `regional1.company2.com` and `regional2` is in `regional2.company3.com`. Modify the `protectedDomains` section of the `nms-ui.properties` file on `global1`, `regional1` and `regional2` as follows:

```
com.hp.nms.ui.sso.protectedDomains=regional1.company1.com, regional2.company2.com, regional3.company3.com
```

- Save your changes.
- Run the following command sequence on `global1`, `regional1`, and `regional2`:
 - On Windows: `%nnminstalldir%\bin\nnmsso.ovpl -reload`
 - On Linux: `/opt/OV/bin/nnmsso.ovpl -reload`

Note: There are no manual configuration steps to perform to enable single sign-on in an application failover configuration. For example, if you plan to configure single sign-on in an application failover configuration, NNMi replicates the above changes from the active NNMi management server to the standby NNMi management server.

Configuring Forwarding Filters on the Regional Managers

In this example, `global1` communicates with both `regional1` and `regional2`. To control the node object data you want the global manager, `global1`, to receive from regional managers `regional1` and `regional2`, you must configure forwarding filters on both `regional1` and `regional2`.

Configuring a Forwarding Filter to Limit Forwarded Nodes

The example creates a node group to enable `regional1` to only forward node information for Procurve Model 3500yl switches to `global1`. To create a new node group and set these limits, complete the following steps:

1. From `regional1`'s **Configuration** workspace in the NNMi console, click **Node Groups**.
2. Click **New**.

Note: Although this example explains how to create a new node filter, then use it to create a forwarding filter from `regional1` and `regional2`, you can use any of existing filters to set up forwarding filters from a regional NNMi management server to a global NNMi management server.

Tip: 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.

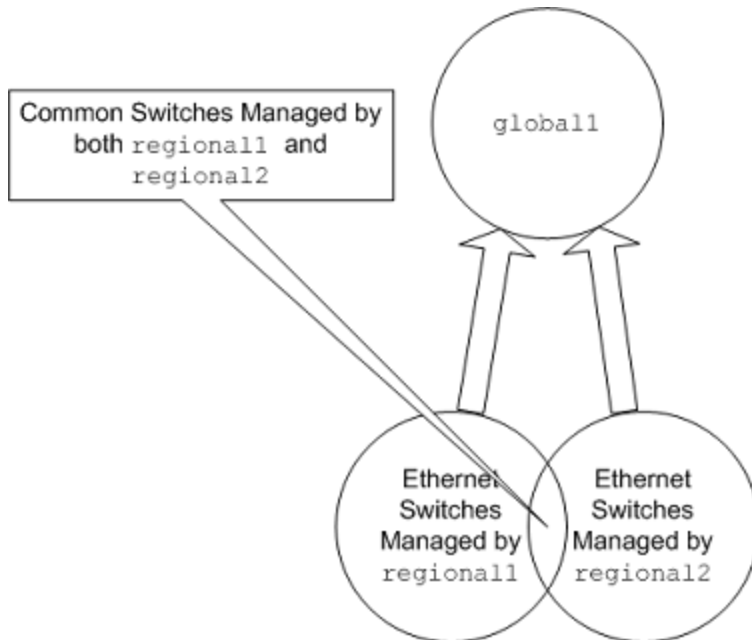
3. Click the **Device Filters** tab. Type `global1` as the filter name and make any notes you need about the filter you are creating in the notes field.
4. Click the **New** icon to open a Node Device Filter form.
5. Using the pull-down, select the Switch Router Device Category, the Hewlett-Packard Device Vendor, and the HP Procurve 3500 Fixed-port Switch Device Family.
6. Using the pull-down, click **Quick Find** to open a Device Profile form.
7. Find and select the profile for the HP Procurve 3500yl Switch; then click **OK**.
8. Click **Save and Close** for each configuration form.
9. To test this filter, select **global1**.
10. Using the pull-down, click **Show Members**.
11. Notice that NNMi discovered 1 HP 3500yl switch already. This shows you that the filter you created is finding the specific switch models you configured it for. The next step is to configure the forwarding filter using this node filter you just created.
12. From `regional1`'s **Configuration** workspace in the NNMi console, click **Global Network Management**.
13. Click the **Forwarding Filter** tab.
14. Click **Quick Find**.
15. Select the **global1** filter; then click **OK**.
16. Click **Save and Close**.

This completes the task of setting up a forwarding filter on `regional1`. After you complete [step 1](#) through [step 16](#) for `regional2`, you are ready to connect `global1` to `regional1` and `regional2` as described in ["Connecting a Global Manager with a Regional Manager" on the next page](#).

Connecting a Global Manager with a Regional Manager

In this example, both `regional1` and `regional2` manage several common switches.

To forward this common switch information to `global1` from `regional1`, you need to set up the required connection.



To make that happen you must connect `global1` to `regional1` before connecting it to `regional2`. By using that 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: HPE recommends 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. See *Clock Synchronization Issues* in the NNMi help for more information.

Note: NNMi displays a warning message if there is a connection problem with a regional manager, such as a server clock synchronization problem.

2. Set up a connection from `global1` to `regional1`.
 - a. From the `global1` NNMi console, click **Global Network Management** in the **Configuration** workspace.
 - b. Click **Regional Manager Connections**.
 - c. Click the **New** icon to create a new regional manager.
 - d. Add the name and description information for `regional1`.

- e. Click the **Connection** Tab.
- f. Click the **New** icon.
- g. Add the connection information for regional1

Note: See **Help->Using the Regional Manager Connection Form** in the NNMi help for specific information about the completing this form.

- h. Click **Save and Close** in each configuration form to save your changes.
3. Complete [step a](#) through [step g](#) to establish a connection from global1 to regional2.

Determining the Connection States from global1 to regional1 and regional2

To check the connection states from global1 to regional1 and regional2, complete the following steps:

1. From the global1NNMi console, click **Global Network Management** in the **Configuration** workspace.
2. Click the **Regional Managers Connections** tab.
3. Check the status of regional1 and regional2 by checking their connection states. Notice that the connection states are shown as Connected, which means they are functioning properly.

See *Determine the State of the Connection to a Regional Manager* in the NNMi help for more information.

Do not continue to the next section until NNMi completes discovery. See *Checking Discovery Progress* in the HPE Network Node Manager i Software Interactive Installation Guide for more information.

Reviewing global1 Inventory

Do not complete this section until NNMi completes discovery. See *Checking Discovery Progress* in the HPE Network Node Manager i Software Interactive Installation Guide for more information.

To view the node information regional1 forwarded to global1, complete the following steps:

1. From the global1NNMi console, navigate to the **Nodes by Management Server** form located in the **Inventory** workspace.
2. Assume that regional1 passed information about switch procurve1.x.y.z to global1. After selecting **regional1**, the inventory might look as follows:

Complete [step 1](#) through [step 2](#) to look at the device inventory passed to global1 from other connected regional managers.

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 regional managers.

This example assumes that global1 still has active subscriptions to regional1.

To disconnect communication between global1 and regional1, follow these steps:

1. From the global1NNMi console, click **Global Network Management** in the **Configuration** workspace.
2. Click **Regional Manager Connections**.
3. Check to make sure the status is Connected. If the status is not Connected, diagnose the problem using information from the *Troubleshoot Global Network Management* topic in the NNMi help before continuing.
4. Select regional1, then click the **Open** icon.
5. Click **Connection**, select **regional1.x.y.z**, then click the **Delete** icon.
6. Click **Save and Close**.
7. In the **Regional Manager Connections** tab, note the **Name** attribute value for regional1 (case-sensitive). You need this text string for the RemoteNNMiServerName variable in a later step.
8. Click **Save and Close**.
9. On global1, at the command line, type the following command:


```
nmnodedelete.ovpl -rm regional1 -u NNMiadminUserName -p NNMiadminPassword
```
10. These commands remove the node records from global1 that regional1 forwarded to it. The commands also close incidents associated with the nodes forwarded to global1 from regional1. For detailed information, see *Disconnect Communication with a Regional Manager* in the 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. Select the **Regional Manager Connections** tab.
 - d. Select regional1, then click the Delete icon.
 - e. Click **Save and Close** to save your deletions.
12. Complete [step 1](#) through [step 11](#) for other regional NNMi management servers, such as regional2, that are connected to global1.

Discovery and Data Synchronization

As network administrators add, delete, or modify devices on a network, regional servers, such as regional1 and regional2, discover those changes and update global servers, such as global1 in the example in this chapter, regional1 and regional2 also notify global1 of changes that administrators make to the management mode of a node it manages.

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.

global1 synchronizes with regional1 and regional2 each time regional1 or regional2 completes a discovery. NNMi uses FDB (Forwarding Database) 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.

Note: Changes to user-modified or application-modified attributes are not updated on the global during a synchronization.

The `Rediscovery Interval` is adjustable on each regional, and can make a difference in the discovery accuracy between `global1` and the regional managers. The shorter the `Rediscovery Interval`, the more accurate the discovery, and the more NNMi-generated network traffic. The longer the `Rediscovery Interval`, the less accurate the discovery, and the less NNMi-generated network traffic. This means that the larger your network grows, the less frequently you might want to rediscover. To set the `Rediscovery Interval`, do the following steps:

1. From the `regional1` or `regional2` NNMi console, click **Discovery Configuration** in the **Configuration** workspace.
2. Adjust the `Rediscovery Interval` according to your how often you want the regionals to initiate a discovery. The global will initiate a discovery immediately after a regional completes a discovery.
3. Click **Save and Close**.

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 in the NNMi console using the global manager's **Custom Attribute Replication** tab (within **Global Network Management** configuration).

Note: NNMi replicates custom attributes for unnumbered interfaces without any user configuration or input. See the NNMi help for more information.

In addition, you can use the `nnmgnmattrcfg.ovpl` command line interface tool to do the following:

- Add attributes to be replicated
- Remove attributes from being replicated
- Add attributes to be replicated using a file for bulk operations
- Remove attributes from being replicated using a file for bulk operations

See the `nnmgnmattrcfg.ovpl` reference page, or the Linux manpage, for more information.

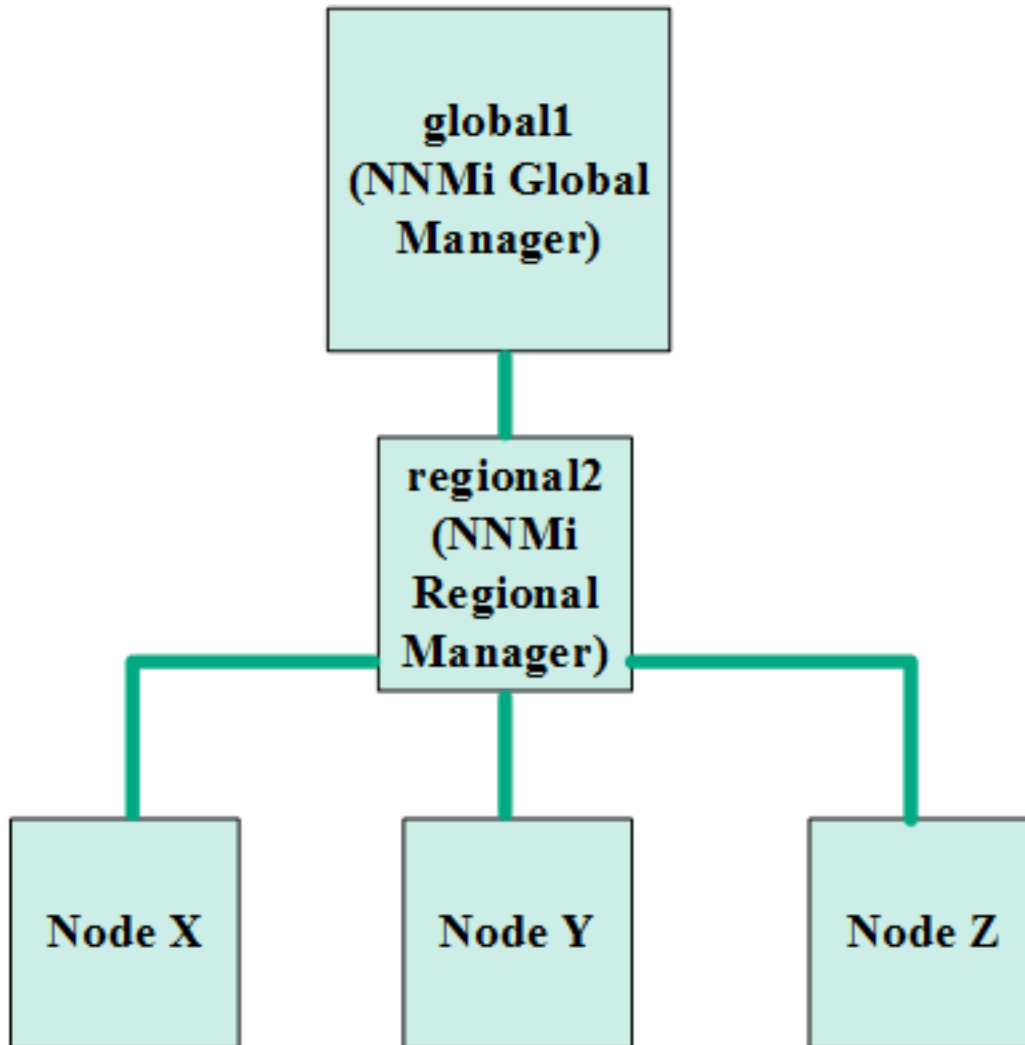
Status Poll or Configuration Poll a Device

This example assumes the following (see the following diagram):

- Regional NNMi management server `regional2` discovers and manages Node X
- Global NNMi management server `global1` connects with regional NNMi management server `regional2`.

Status Poll or Configuration Poll a Node

Global Network Management



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. Request a status poll of Node X using the **Actions > Status Poll** menu item.
4. NNMi management server global1 requests a status poll from regional NNMi management server regional2 and shows the results on your screen. It does not matter if you initiate the status poll request from either global1 or regional2. You see the same status poll results.

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. Request a configuration poll of Node X using the **Actions > Configuration Poll** menu item.
4. NNMi management server `global1` requests a configuration poll from regional NNMi management server `regional2` and shows the results on your screen. It does not matter if you initiate the configuration poll request from either `global1` or `regional2`. You see the same configuration poll results.

Determining Device Status and NNMi Incident Generation using a Global Manager

NNMi management server `global1` listens for state changes coming from regional managers `regional1` and `regional2` and updates the states in its local database.

The NNMi `StatePoller` services on NNMi management servers `regional1` and `regional2` calculate 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`.

After you change the management mode of a node being managed by `regional1`, you see that management mode change on `global1` as well. As network administrators add, remove, or modify network equipment being managed by `regional1` or `regional2`, `regional1` or `regional2` updates `global1` of these network device changes.

`global1` generates incidents using its own causal engine and topology, including the node object data forwarded to it by `regional1` 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` or `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, `regional1` does not forward traps to `global1`. To do this, you must configure `regional1` to forward specific traps to `global1`. HPE recommends you only configure regional managers to forward low-volume, important traps to avoid excessive burden on the global manager. NNMi drops forwarded traps if the forwarded traps result in a `TrapStorm` incident. See the `TrapStorm` Management Event details in the NNMi console.

Configuring Application Failover for Global Network Management

You can configure both global and regional managers to use application failover. The global or regional manager automatically detects and connects to the active system.

To configure `global1` to recognize the application failover do the following:

1. From the `global1` NNMi console, click **Global Network Management** in the **Configuration** workspace. This example assumes the following:
 - `regional1` is configured for application failover
 - `regional1_backup` is configured as the secondary server
2. Click **Regional Manager Connections**.

3. Select `regional1`, then click the **Open** icon.
4. Click the **New** icon.
5. Add the **Hostname**, **HTTP or HTTPS Port**, **User Name**, and **Ordering** value. Set the ordering value to a value greater than the `regional1` value.
6. Click **Save and Close** in each configuration form to save your changes.

If a regional manager fails, the global manager does the following:

- a. It contacts the primary.
- b. If the primary does not respond, it contacts the secondary.

If the global system detects that the active system is not responding, it tries to reconnect starting with the lowest order number.

Verify the Global Network Management Configuration

Perform the tasks listed in this topic to troubleshoot global network management configuration.

Check Clock Synchronization Status

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, for example, Linux tool Network Time Protocol Daemon (NTPD) or one of the available Windows operating system tools.

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

```
NNMi is not connected to 1 Regional Manager(s). See Help ? System Information, Global Network Management.
```

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

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

Perhaps the clocks have drifted apart and need to be resynchronized. Check the `nm.0.0.log` file on the Global Manager for the following message:

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

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

```
NNMi is not connected to 1 Regional Manager(s). See Help ? System Information, Global Network Management.
```

View System Information

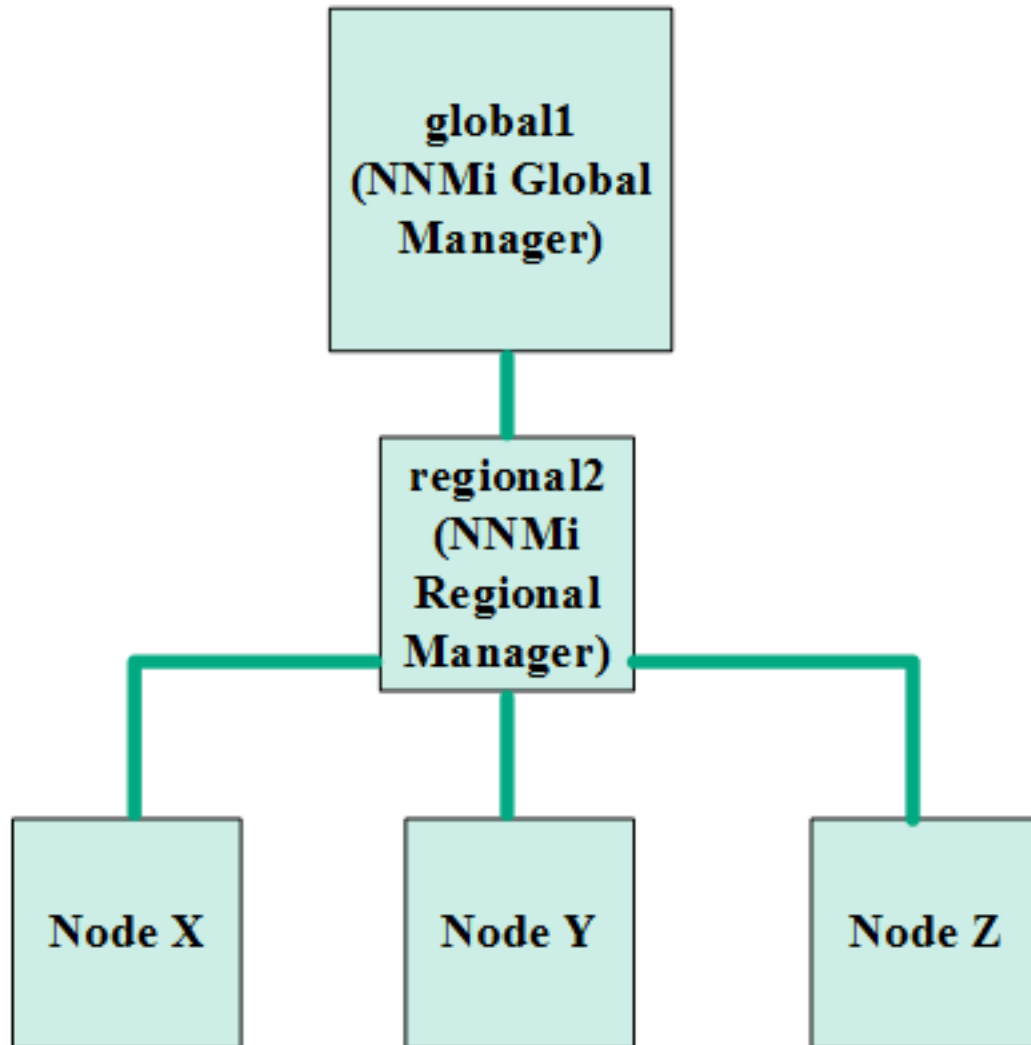
On the global manager, log on to the NNMi console, select **Help > System Information**, and then click the **Global Network Management** tab to view information about your global network management connections.

Synchronize Regional Manager Discovery from a Global Manager

If you notice an information inconsistency between `global1` and `regional2`, run the `nnmnode rediscover.ovpl` script from `global1`, causing `global1` and `regional2` to synchronize. This also results in the `regional2` updating `global1` with any new discovery results.

This example uses the network shown in the following diagram.

Global Network Management



Run the following command to synchronize nodes X, Y, and Z with `global1`:

```
nnmnode rediscover.ovpl -u username -p password -rm regional2.
```

You can use the `-fullsync` flag with the `nnmnode rediscover.ovpl` command to synchronize all polled object states and status (although this takes more time and causes a greater load on the systems). For more information, see the `nnmnode rediscover.ovpl` reference page, or the Linux manpage.

- NNMi automatically resynchronizes topology, state, and status following a manual resynchronization.
- Avoid stopping NNMi during the resynchronization. To help ensure resynchronization has completed, NNMi should remain running for several hours following the manual resynchronization. The actual time required depends on the number of nodes and the volume of state changes and trap data received while performing the resynchronization.
- If NNMi must be stopped before the resynchronization is finished, the resynchronization should be run again and allowed to complete.
- To perform a manual resynchronization of the entire management server, run:
`nnmnode rediscover.ovpl -all -fullsync`

Recover a Destroyed Database on global1

If you take global1 out of service and need to restore its database, you face several scenarios:

1. If you restore global1's database successfully, regional1 and regional2 synchronize their cached information with global1. There are no manual steps to perform after bringing global1 back online.
2. If global1 is out of service for an extended period of time, step 1 might not work successfully. To remedy this, run the `nnmnode rediscover.ovpl` script on global1 to initiate a new discovery on global1, regional1 and regional2. In this case you could run status polls on key devices to more quickly get updated status information.
3. If you cannot recover global1's database then you should submit a support call to clear out the old global1 data from the regional1 and regional2 databases using the `nnmsubscription.ovpl` script.

Global Network Management and NNM iSPIs or Third-Party Integrations

Each NNM iSPI or third-party integration has its own unique deployment guideline. For the examples in this chapter, you can deploy some NNM iSPIs on regional1 only, global1 only, or on both regional1 and global1. For other NNM iSPIs or third-party integrations, you must have them installed on both regional1 and global1. See the documentation for the NNM iSPI or third-party integration for more information.

Global Network Management and Address Translation Protocols

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. See ["Managing Overlapping IP Addresses in NAT Environments" on page 344](#). See also the NNMi help.

Configuring NNMi Advanced for IPv6

You must purchase and install an NNMi Advanced, NNMi Premium or NNMi Ultimate license to use the IPv6 management feature.

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. See the NNMi help for more information about IPv6 information displayed in the NNMi console.

This chapter contains the following topics:

- ["Feature Description" below](#)
- ["Prerequisites" on the next page](#)
- ["Licensing" on page 396](#)
- ["Supported Configuration" on page 396](#)
- ["Installing NNMi" on page 397](#)
- ["Deactivating IPv6 Features" on page 398](#)
- ["Reactivating IPv6 Features" on page 399](#)

Feature Description

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 address) for dual-stacked devices. Use the NNMi console to set the SNMP management address preference to IPv4 or IPv6 using **Communication Configuration** located in the **Configuration** workspace.
- Native ICMPv6 communication for IPv6 Address fault monitoring.
- Seeded device discovery using an IPv6 address or hostname
- Automatic IPv6 device discovery using IPv6 Layer 3 neighbor discovery hints
- Automatic IPv6 device discovery using layer 2 neighbor discovery hints using LLDP (Link Layer Discovery Protocol) 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
- SDK Web-services support for IPv6 inventory and incidents
- NNM iSPI Performance for Metrics support for IPv6 interfaces

The NNMi IPv6 management feature excludes the following:

- Discovery of IPv6 subnet connections
- 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

Prerequisites

Review the NNMi Deployment Reference, NNMi Release Notes, and *NNMi Support Matrix* for details on management server specifications and NNMi installation.

To use native IPv6 communication, the NNMi management server must be a dual-stacked system, meaning that it communicates using both IPv4 and IPv6.

Note: If you have IPv6 discovery configured on NNMi, and are using the Universal CMDB (UCMDB) integration, the UCMDB HPE Discovery and Dependency mapping (DDM) import task fails. You need to disable IPv6 discovery to use the UCMDB integration with NNMi.

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 global unicast address or a unique local unicast address configured on at least one network interface that is connected to the IPv6 network you want to 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 having IPv6 addresses.

The DNS server used by the NNMi management server must resolve hostnames to and from IPv6 addresses. For example, it must be able to resolve to and from an AAAA DNS record. That means the DNS server must map a hostname 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 nor display DNS hostnames for nodes using IPv6 addresses.

Licensing

You must purchase and install an NNMi Advanced, NNMi Premium or NNMi Ultimate license to use the IPv6 management feature. For information about obtaining and installing an NNMi license, see ["Apply Licenses" on page 241](#).

The NNMi product includes a temporary Instant-On license password. This is a temporary, but valid NNMi Advanced license. You should obtain and install a permanent license password as soon as possible.

Supported Configuration

See the NNMi Support Matrix for additional information about the supported operating system configurations for NNMi.

Management Server

The following table shows the capabilities of both the IPv4-only and dual-stacked NNMi management server.

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	Supported	Supported

Management Server Capabilities, continued

Feature/Capability	IPv4-Only	Dual-Stack
using SNMP		
IPv4 Address Status using ICMP	Supported	Supported
IPv6 Address Status using ICMPv6	Not Supported	Supported
IPv6-only Managed Node	Not Supported	Supported
Discovery using IPv6 Seed	Not Supported	Supported
IPv6 Address and Subnet Inventory	Not Supported	Supported
Interface Status and Performance using SNMP	Not Supported	Supported
IPv6 Address Status using ICMPv6	Not Supported	Supported
IPv4-only Managed Node	Supported	Supported
Node Discovery using IPv4 Seed	Supported	Supported
Node Discovery using IPv4 Seed	Supported	Supported
Interface Status and Performance using SNMP	Supported	Supported
Interface Status and Performance using SNMP	Supported	Supported
IPv4 Address and Subnet Inventory	Supported	Supported

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

Installing NNMi

During NNMi installation, the installation script activates IPv6 features; however, you can manually deactivate these IPv6 features, if desired, by editing the `nms-jboss.properties` file.

You can later reactivate IPv6 features after they have been deactivated. See ["Deactivating IPv6 Features" below](#) and ["Reactivating IPv6 Features" on the next page](#) for more information.

Deactivating IPv6 Features

You can administratively disable IPv6 features by doing the following:

1. Open the `nms-jboss.properties` file. Look in the following location:

Windows: `%NNM_PROPS%\nms-jboss.properties`

Linux: `$NNM_PROPS/nms-jboss.properties`

Note: 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:
 - c. `java.net.preferIPv4Stack=false`
 - d. 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. Restart the NNMi management server.
 - a. Run the `ovstop` command on the NNMi management server.
 - b. Run the `ovstart` command on the NNMi management server.

Note: 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 ["Maintenance Mode" on page 177](#) for more information.

5. Check the NNMi processes using the following command:

`ovstatus -v ovjboss`

For information about changing the NNMi license, see ["Licensing" on page 396](#).

IPv6 Monitoring Following Deactivation

If IPv6 management or IPv6 communication becomes completely disabled, the StatePoller service immediately stops monitoring IPv6 addresses with ICMPv6. NNMi sets the IP address state of these addresses to Not Polled. If you select an address, then use the **Actions > Monitoring Settings** for this address, NNMi displays `Fault ICMP Polling enabled: false` even though the associated Monitoring Configuration rule has the IP Address Fault Polling enabled.

IPv6 Inventory Following Deactivation

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. NNMi does not remove non-SNMP IPv6 nodes. You must manually delete IPv6 nodes from the NNMi inventory.
- NNMi Advanced *only*. Your NNMi Advanced license expired or someone removed the license. NNMi begins using the NNMi basic license, and the basic license has enough capacity to continue managing all of the discovered nodes.
NNMi immediately removes all of the non-SNMP IPv6 nodes from its inventory. NNMi rediscovers all of the SNMP nodes and removes all of the IPv6 data.
- NNMi Advanced *only*. Your NNMi Advanced license expired or someone removed the license. NNMi begins using the NNMi basic license, and the basic license does not have enough capacity to continue managing all of the discovered nodes. NNMi immediately removes all non-SNMP IPv6 nodes.

Known Issues When Cleaning Up IPv6 Inventory

You could experience leftover IPv6 inventory in the following situation:

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 these remaining nodes, you must fix the communication problem, then use the **Actions > Configuration Poll** command located in the NNMi console to obtain configuration information from these nodes. For native IPv6 nodes, delete the node directly from the NNMi console.

Reactivating IPv6 Features

Note: Features requiring IPv6 communication, such as the discovery and of IPv6 only devices and the monitoring of IPv6 address status, require an NNMi management server to have an IPv6 global unicast address configured and operational.

The following procedure explains how to reactive IPv6 features after they have been deactivated.

1. Edit the `nms-jboss.properties` file. Look in the following location:
Windows: `%NNM_PROPS%\nms-jboss.properties`

Linux: \$NNM_PROPS/nms-jboss.properties

Note: NNMi provides a complete description of each property, showing them as comments in the nms-jboss.properties file.

2. Locate the text that begins with # Enable NNMi IPv6 Management.
3. To enable IPv6 communication in NNMi, un-comment the property:

```
java.net.preferIPv4Stack=false
```

Note: To un-comment a property, remove the #! characters from the beginning of a line.

4. Locate the text that begins with # Enable NNMi IPv6 Management.
5. To enable overall IPv6 management in NNMi, un-comment the property:

```
com.hp.nnm.enableIPv6Mgmt=true
```

6. Save and close the nms-jboss.properties file.
7. Restart the NNMi management server.
 - a. Run the **ovstop** command on the NNMi management server.
 - b. Run the **ovstart** command on the NNMi management server.

Note: 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 "[Maintenance Mode](#)" on page 177 for more information.

8. Check the NNMi processes using the following command:

```
ovstatus -v ovjboss
```

Successful startup should look something like the following:

```
object manager name: ovjboss
state:                RUNNING
PID:                  <Process ID #>
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	
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

9. After you reactivate IPv6, NNMi views immediately include the IPv6 inventory for newly discovered nodes. During the next discovery cycle, NNMi views show the IPv6 inventory associated with previously discovered nodes.
10. Optionally set the SNMP management address preference for dual-stacked managed nodes. Dual-stacked managed nodes are those 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. Restart the NNMi management server:
 - Run the **ovstop** command on the NNMi management server.
 - Run the **ovstart** command on the NNMi management server.

Note: 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 ["Maintenance Mode" on page 177](#) for more information.

To speed things up, select nodes that you know are dual-stack nodes, and then use the **Actions > Configuration Poll** command located in the NNMi console. You can also use the `nnmnodediscover.ovpl` script to add nodes to the NNMi discovery queue. See the `nnmnodediscover.ovpl` reference page, or the Linux manpage, for more information.

After you enable IPv6 communication on the NNMi management server, NNMi begins monitoring nodes for IPv6 address faults using ICMPv6.

Chapter 8: NNMi Security

This chapter contains the following topics:

- "Configuring SSL Communications for Web Access and RMI Communications" below
- "Allowing Non-Root Linux Users to Start and Stop NNMi" on the next page
- "Providing a Password for Embedded Database Tools" on the next page
- "Configuring NNMi to Enable or Disable SSLv3 Ciphers" on page 405
- "Configuring NNMi Ciphers" on page 407
- "NNMi Data Encryption" on page 407

Configuring SSL Communications for Web Access and RMI Communications

NNMi includes a suite of default ciphers that are used in configuring Secure Sockets Layer (SSL) in Web access and Java Remote Method Invocation (RMI) communications. The ciphers are listed in the `nms-jboss.properties` file.

Caution: Adding or removing ciphers from the cipher list without the approval of HPE is not supported; doing so may cause damage to the product or cause the product to become inoperable.

Requirement for New NNMi 10.30 Installations

New installations of NNMi support only TLS v1.2 protocol by default. However, to be able to discover and monitor ESXi 5.1 hypervisors, NNMi is required to use the TLSv1 cryptographic protocol.

To configure NNMi to support the TLSv1 cryptographic protocol for device communication:

Note: This procedure enables NNMi to use less secure cryptographic protocols that are not FIPS 140-2-certified. This is a global change and may reduce the security of the product.

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. Update the `com.hp.ov.nms.ssl.PROTOCOLS` property to include the value `TLSv1`.
If the property does not exist, add the following line:
`com.hp.ov.nms.ssl.PROTOCOLS=TLSv1.2,TLSv1.1,TLSv1`
4. Configure NNMi to allow protocols and algorithms that are not FIPS-certified:

- a. On the NNMi management server, go to the following directory:
 - o *On Windows:* %nnminstalldir%\newconfig\HPNmsServStgs\Windows
 - o *On Linux:* /opt/OV/newconfig/HPNmsServStgs/Linux
 - b. Copy the java.security file, and then place the copied file in the following directory:
 - o *On Windows:* %nnmdatadir%\conf\nnm
 - o *On Linux:* /var/opt/OV/conf/nnm
5. Restart the NNMi processes by running the following commands:
- *On Windows:*
 - i. %nnminstalldir%\bin\ovstop -c
 - ii. %nnminstalldir%\bin\ovstart -c
 - *On Linux:*
 - i. /opt/OV/bin/ovstop -c
 - ii. /opt/OV/bin/ovstart -c

Allowing Non-Root Linux Users to Start and Stop NNMi

Note: If the /opt/OV directory is on a partition with the nosuid option set, the non-root user feature is not available. See /etc/fstab to determine if the partition is configured with the nosuid option set.

NNMi provides a way to allow non-root Linux users to start and stop NNMi. Do the following:

1. As root, edit the following file:


```
$NnmDataDir/shared/nnm/conf/ovstart.allow
```
2. Include the non-root users (one per line) that you want to be able to start and stop NNMi.
3. Save your changes.

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. See "[Maintenance Mode](#)" on page 177 for more information.

Providing a Password for Embedded Database Tools

To run embedded database tools (such as psql), NNMi requires a password. NNMi provides a default password, which the user should change using the nnmchangeembdbpw.ovpl script.

Note: You must be logged in as administrator on Windows systems or root on Linux systems to run the nnmchangeembdbpw.ovpl script. For more information, see the nnmchangeembdbpw.ovpl reference page, or the Linux manpage

If you have configured NNMI in an High Availability (HA) environment, run the `nnmchangeembdbpw.ovpl` script on the Primary Cluster Node only.

On the Primary Cluster Node only:

1. Place the Primary Cluster Node into maintenance mode.
See "[Maintenance Mode](#)" on page 177 for more information about placing nodes in maintenance mode.
2. Stop all NNMI processes:
Windows: `%NNM_BIN%\ovstop -c`
Linux: `$NNM_BIN/ovstop -c`
3. Restart `nnmsdbmgr`:
Windows: `%NNM_BIN%\ovstart nnmsdbmgr`
Linux: `$NNM_BIN/ovstart nnmsdbmgr`
4. To change the embedded database password, run the `nnmchangeembdbpw.ovpl` script.
Windows: `%NNM_BIN%\nnmchangeembdbpw.ovpl`
Linux: `$NNM_BIN/nnmchangeembdbpw.ovpl`
5. To ensure the change is copied to the replication directory, so it can be copied to the Secondary Cluster Node, run the `nnmdatareplication.ovpl` script:
Windows: `%NNM_DATA%\misc\nnm\ha\nnmdatareplication.ovpl NNM`
Linux: `$NNM_DATA/misc/nnm/ha/nnmdatareplication.ovpl NNM`
6. Restart all NNMI processes:
Windows: `%NNM_BIN%\ovstart`
Linux: `$NNM_BIN/ovstart`
7. Take the Primary Cluster Node out of Maintenance Mode.
8. Fail over to the Secondary Cluster Node.

Note: The Secondary Cluster Node must NOT be in Maintenance Mode in order to have the Postgres password replicated.

The application automatically copies the password to the Secondary Cluster Node when the NNMI Resource Group is started on this node.

Configuring NNMI to Enable or Disable SSLv3 Ciphers

You can modify the NNMI list of ciphers. However, ensure that the original information is preserved by copying the properties file discussed in this section to a different directory. NNMI disables SSLv3 ciphers by default. You might need to enable SSLv3 ciphers to resolve web browser communication issues. For example, you might receive a connection error similar to one of the following:

- Secure Connection Failed
- This page can't be displayed

If you are also using NNM iSPI software that resides on the NNMi management server and you enable SSLv3 ciphers for NNMi, you must also enable SSLv3 for each iSPI. See the Deployment Reference for each corresponding NNM iSPI for information about enabling and disabling SSLv3.

When making file changes under High Availability (HA), the location of the server.properties file that you need to update is: <Shared_Disk>/NNM/dataDir/nmsas/NNM/server.properties.

To configure NNMi to enable SSLv3 ciphers:

1. Open the following file:

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

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

2. Edit the following line:

```
com.hp.ov.nms.ssl.PROTOCOLS = SSLv2Hello,TLSv1,TLSv1.1,TLSv1.2
```

to include SSLv3. For example:

```
com.hp.ov.nms.ssl.PROTOCOLS = SSLv2Hello,TLSv1,TLSv1.1,TLSv1.2,SSLv3
```

Note: You can remove any protocols contained in this line.

3. Save the file.

Note: If you are also enabling SSLv3 for one or more iSPIs, make those changes before stopping and starting the NNMi management server as described in the next steps.

4. Stop the NNMi management server:

Run the ovstop command on the NNMi management server.

5. Re-start the NNMi management server:

Run the ovstart command on the NNMi management server.

To disable the SSLv3 ciphers after they have been enabled:

1. Open the following file:

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

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

2. Edit the following line:

```
com.hp.ov.nms.ssl.PROTOCOLS = SSLv2Hello,TLSv1,TLSv1.1,TLSv1.2,SSLv3
```

to remove SSLv3. For example:

```
com.hp.ov.nms.ssl.PROTOCOLS = SSLv2Hello,TLSv1,TLSv1.1,TLSv1.2
```

3. Save the file.

Note: If you are also disabling SSLv3 for one or more iSPIs after it has been enabled, make those changes before stopping and starting the NNMi management server as described in the next steps.

4. Stop the NNMi management server:

- Run the `ovstop` command on the NNMi management server.
- 5. Re-start the NNMi management server:
 - Run the `ovstart` command on the NNMi management server.

Configuring NNMi Ciphers

For information about the ciphers that NNMi uses, see "Configure the Ciphers Used by the NNMi Web Server" in the *NNMi Hardening Guide*.

NNMi Data Encryption

NNMi incorporates data encryption in many areas of the product. For example:

- Application failover encrypts messages sent between cluster nodes.
- NNMi stores passwords for user accounts in the NNMi database in encrypted form.
- Global Network Management (GNM) encrypts messages sent between regional managers and the global manager.

NNMi uses a method of data encryption that spans several NNMi components. NNMi data encryption supports the following encryption types:

- symmetric encryption—both parties share the same secret key
- asymmetric—public and private key encryption where each side has the other side's public key, but they keep their own private key
- MessageDigest (hash)—one-way encryption (cannot decrypt) where arbitrarily long strings are reduced to fixed size strings

This topic describes the default security configurations for encryption and hashing within NNMi.

- A new installation of NNMi10.30 uses Federal Information Processing Standards (FIPS) 140-2-validated cryptographic module (RSA BSAFE) for encryption and key management.

In an upgraded NNMi environment, FIPS-compliant ciphers and algorithms are automatically used for most password encryption and network communication procedures. However, some legacy ciphers and algorithms do exist in the upgraded environment that do not meet FIPS guidelines.

- During installation, NNMi generates a self-signed certificate using a 2048-bit encryption key, SHA 256, and RSA.

Note: HPE recommends using a CA-signed certificate instead of the self-signed certificate provided by NNMi.

- For local authentication into NNMi, NNMi uses a salted SHA-256 password hash for storing NNMi user passwords.
- For encryption of device passwords stored in the NNMi database, NNMi uses the AES 128 algorithm.

Encryption Configuration Files

The NNMi encryption framework includes a set of files that can be edited to configure encryption settings for your organization. The files are in the following folder:

- Windows: %NnmDataDir%\shared\nnm\conf\crypto
- Linux: \$NnmDataDir/shared/nnm/conf/crypto

Caution: The crypto configuration files are intended for advanced users. Use extreme caution when editing the crypto configuration files. Improper editing of these files cause serious issues. For example, any changes to the encryption parameters for application failover causes application failover to no longer function. Likewise, changes to system and database password encryption settings causes NNMI to no longer start. See the following sections for procedures to follow when changing crypto configurations for different NNMI subsystems.

Text Blocks in the Crypto Configuration Files

The crypto configuration files include the following text blocks:

<allowed>

The <allowed> block defines the types of providers, algorithms, and minimum key lengths that are allowed to be used elsewhere in the crypto configuration files.

Note: If you attempt to use an algorithm or key length that is not allowed, NNMI generates an encryption error.

Tip: A provider is a vendor (or entity) that provides implementations of cryptographic algorithms.

The algorithms listed in the crypto configuration files are associated with the providers listed in those files.

<default>

The <default> block lists default settings used for all supported components. For example, the <default> block lists a one symmetric algorithm, one asymmetric algorithm, and one digest. If there is a component block defined for a given component, that component uses the algorithm specified in its component block (in other words, the component block definition overrides the <default> block). Otherwise, a component will request the default algorithm (from the <default> block) for the specific type of encryption used by that component.

Each component uses only one type of encryption (symmetric, asymmetric, or digest). For example, application failover uses only symmetric encryption, so specifying an asymmetric or digest algorithm in an application failover component block would be ineffective and unnecessary.

Note: A key size listed in a default block or component block must be at least the size listed in the <allowed> block (but it can be greater, if desired). For example, if the <allowed> block includes AES-128, then AES-192 is also valid. However, if the <allowed> block specifies AES-192, AES-128 is not valid.

Encryption and Application Failover

To make encryption configuration changes for application failover (for example, changing an encryption algorithm or key length) do the following:

1. Stop NNMi and nmcluster processes by running the ovstop command on both nodes. Note that when you use the ovstop command on an NNMi management server configured for application failover, NNMi automatically runs the following command:

```
nmcluster -disable -shutdown
```

2. Edit the nmcluster-crypto-config.xml file as desired.

Note: Application failover uses only symmetric encryption, so adding asymmetric or digest does not have any effect, and removing symmetric causes a failure.

3. Save your changes to the nmcluster-crypto-config.xml file.
4. Remove the old key file.

Tip: The file location is defined in the nmcluster-crypto-config.xml file.

5. Generate a new key file by running the following command:

```
nmcluster -genkey
```

6. Copy the edited nmcluster-crypto-config.xml file and the new key file to the other node in the cluster (in the same folders).

Now the nmcluster-crypto-config.xml file, which defines the encryption algorithms and keys, is the same on both nodes. Also, the key itself is the same on both nodes.

7. Start the cluster again by running nmcluster on the active and standby nodes:

```
Run nmcluster -daemon on the active node
```

Note: Wait until the node becomes active

```
Run nmcluster -daemon on the standby node
```

Note: If you do not remove the old key file, you might receive an error similar to the following:

```
Warning: Generating a new encryption key will require the NNMi Cluster to be shutdown.
```

```
Do you wish to continue (y/n)?
```

```
y
```

```
Error: The attempt to generate a new encryption key failed.
```

```
The most likely cause is that the keysize was increased and the current key is invalid.
```

```
Please remove the existing key and try again.
```

Encryption and User Account Passwords

Note: This information does not apply to Lightweight Directory Access Protocol (LDAP) or Common

Access Card (CAC) accounts.

NNMi user accounts created using the NNMi console are stored in the NNMi database. The passwords for these users are hashed and stored in the database.

When users sign into the NNMi console, or use a command line interface (CLI) tool, the password that they provide is hashed and compared to the hashed value stored in the database. If the user provides the correct password, these two hashed strings match, and the user is authenticated.

Earlier versions of NNMi (9.x) used encryption algorithms for hashing user passwords, which are now considered outdated. NNMi uses a stronger algorithm for user account passwords. However, since hashes are one-way encryption, it is not possible to decrypt and then re-encrypt the user passwords during and upgrade from NNMi 9.x to 10.x.

On 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 successfully logs on, the password they provided is automatically re-encrypted using the new hash algorithm specified in the crypto configuration files.

This means all passwords are updated to the new algorithm slowly over time, as each user logs in for the first time after upgrade. The same is true if the crypto configuration is changed in the future. User passwords are upgraded to the new hash algorithm on the next successful logon.

- Upgrading user passwords depends on the presence of the earlier legacy algorithm (for example, MD5) listed in the <allowed> block. Therefore, keep the earlier legacy algorithm listed in the <allowed> block until all passwords have been migrated.
- Without the presence of the earlier legacy algorithm in the <allowed> block, the existing passwords hashed in the database are not able to be re-hashed. Therefore, associated users are not be able to log on, and NNMi is not able to re-encrypt passwords using the new algorithm.
- If the earlier legacy algorithm has been removed from the <allowed> block, the administrator must either delete and recreate the users affected, or reset the respective passwords for users whose passwords were encrypted with earlier legacy algorithms.

Use the following command to determine whether a user's password is using the algorithm listed in the crypto configuration file, or the user's password is encrypted with earlier legacy algorithms no longer specified in the crypto configuration file:

```
nnmsecurity.ovpl -listUserAccounts legacy
```

See the `nnmsecurity.ovpl` reference page, or the Linux manpage, for more information.

Chapter 9: Use HPE Operations Bridge Reporter to View Reports

To perform an effective performance analysis of the network with the data collected by NNMi, you can use HPE Operations Bridge Reporter (OBR).

Prerequisites

- NPS is already installed and configured.
- NPS and OBR must not be installed on the same system.
- NNMi and OBR must not be installed on the same system.
- If NNMi and NPS are installed on the same system, you must complete these additional tasks:
 - a. Go to the following directory on the NNMi management server:
 - *Windows:* %nnmdatadir%\shared
 - *Linux:* /var/opt/OV/shared
 - b. Manually share the perfSpi directory with the user NpsUsr.

Note: NpsUsr is created during the installation of NPS.

Configure NNMi to Export Data to OBR

To enable OBR to use the data collected by NNMi:

1. Log on to the NNMi management server as root or administrator.
2. Open the following file with a text editor:
 - *On Windows:* %nnmdatadir%\shared\perfSpi\conf\nmsAdapter.conf
 - *On Linux:* /var/opt/OV/shared/perfSpi/conf/nmsAdapter.conf
3. Add the following line at the end of the file:
`exportToSHR:true`
4. Save the file.
5. Go to the following directory:
 - *On Windows:* %nnminstallldir%\bin
 - *On Linux:* /opt/OV/bin
6. Restart NNMi by running the following commands:

- *On Windows*
 - i. **ovstop -c**
 - ii. **ovstart -c**

- *On Linux*
 - i. **./ovstop -c**
 - ii. **./ovstart -c**

Configure OBR to Use the Data Collected by NNMi

To configure OBR to use the data collected by NNMi, follow the instructions in the *Configuring SHR to Use NNMi Data* section in the *HPE Operations Bridge Reporter 10.00 Administration Guide*.

Use Reports

To use OBR reports to monitor the performance of the network:

1. Launch the OBR BI Launchpad portal in a web browser.
2. Log on to the OBR BI Launchpad portal as administrator.
3. Go to the Document tab.
4. To view reports on network interfaces, expand Network Interface Health, and then double-click a report.
5. To view reports on network devices, expand Network Component Health, and then double-click a report.

Chapter 10: Migrating Performance Insight (OVPI) SNMP Collections of Custom Report Packs to NNMi

If you are using the NNMi Custom Poller feature and HP Performance Insight (OVPI), you can migrate your custom report pack collections in OVPI to NNMi. After the OVPI collections are migrated, it can be used with the NNMi Custom Poller feature.

The NNMi Custom Poller feature enables you to take a proactive approach to network management by using SNMP MIB expressions to specify additional information that NNMi should poll.

A Custom Poller collection defines the information you want to gather (poll) as well as how NNMi reacts to the gathered data. See *Create a Custom Poller Collection* and *Configure Custom Polling* in the NNMi help for more detailed information. See also the *Network Node Manager i Software Step-by-Step Guide to Custom Poller White Paper*.

Note: Use these steps to migrate only the SNMP based collections of the custom report packs in OVPI .

To migrate SNMP collections associated with OVPI custom report packs to NNMi, follow these steps:

1. Identify the collection policies that need to be migrated from OVPI to NNMi.
2. Use the OVPI `collection_manager` tool to export the collection policies within these custom report packs from the OVPI server. For example:

Note: The OVPI server can be either a remote poller or a satellite server that runs the collection.

```
collection_manager -export <file name>
```

See the `collection_manager` reference page for more information.

3. Collect the additional information needed for an NNMi Custom Poller Collection. This information can be supplied to the `nmmigrateovpi.ovpl` command using either of the following methods:

Specify the information for a single teel file as `nmmigrateovpi.ovpl` command line arguments. For example:

```
nmmigrateovpi.ovpl -policyName myPolicy -teelFile /tmp/OVPI/myTeel.TEEL  
-pollInterval 300 -nodeGroup myNodeGroup
```

Specify multiple teel files inside a single policy file using the `-policyFile` argument to the `nmmigrateovpi.ovpl` command. For example:

```
nmmigrateovpi.ovpl -policyFile CP_policy_config.txt -teelDir /tmp/OVPI  
-batchFile generated_CP_commands.txt
```

An exported OVPI collection policy file contains the following columns: `policy_name`, `table_name`, `poll_interval`, `datapipe_name`, `poll_from`, `user_name`, `server_name`, `group`, `group_server`, `desc`

The following table shows how this exported information relates to the required information in the `nmmigrateovpi.ovpl` command:

OVPI collection policy file column	Required fields in <code>nmmigrate.ovpl</code>
<code>policy_name</code>	Policy name
<code>table_name</code>	TEEL file name
<code>poll_interval</code>	Poll interval
<code>group</code>	Node Group

To extract the information for OVPI collection policy file, use the following Linux command:

```
cut -f1,2,3,8 -d',' ' ovpi_collection_policy.txt > CP_policy_config.txt
```

where `ovi_collection_policy.txt` is the name of an example exported OVPI collection policy file and `CP_policy_config.txt` is the name of an example policy file (<policyfile>) that is used as input to the `nmmcustompollerconfig.ovpl` command.

4. Check the content of the exported OVPI collection policy file. When checking the content, note the following:
 - The `table_name` field in the OVPI exported collection policy is assumed to be the same as the TEEL file name, without the `teel` extension. If the TEEL file name is different from the `table_name`, you need to manually edit the file so that the `table_name` matches the TEEL file name.
 - The group name may not correspond to a Node Group in NNMI. If these names do not match, do either of the following:
 - Change this group name to match an NNMI Node Group name when specifying the information to the migration command
 - Create a Node Group to match the exported group name
5. Locate the TEEL files used in the OVPI collection policies.
6. Copy the TEEL files to a temporary location on an NNMI system.
7. Use `nmmigrateovpi.ovpl` to generate the necessary commands that enable you to configure Custom Poller collections using the data contained in the TEEL file or files.

Tip: You can use `nmmigrateovpi.ovpl` to migrate a single TEEL file or multiple TEEL files.

See the `nmmigrateovpi.ovpl` reference page for more information.

Caution: Several fields in the generated Custom Poller configuration commands use default values. If needed, modify these fields to comply with your requirements. See the `nmmigrateovpi.ovpl` reference page for more information.

The following steps describe an example for migrating multiple collections using the `nmmigrateovpi.ovpl` command. This example assumes you have already created and checked the content of the exported OVPI collection policy file as described in the previous procedure.

1. Run the `nmmigrateovpi.ovpl` command:

```
nmmigrateovpi.ovpl -policyFile <file name> -teelDir <directory where the TEEL files are present> [ -batchFile <file name where generated commands are written>]
```

For example:

```
nmmigrateovpi.ovpl -policyFile CP_policy_config.txt -teelDir /tmp/OVPI -batchFile generated_CP_commands.txt
```

2. Use the new batch file with the NNMi Custom Poller configuration command `nnmcustompollerconfig.ovpl` as follows:

```
nmcustompollerconfig.ovpl -batch <batch command file>
```

For example:

```
nmcustompollerconfig.ovpl -batch generated_CP_commands.txt
```

NNMi creates the Custom Poller collections using the configuration information contained in the batch command file.

3. To view these Custom Poller Collections from the NNMi console:
 - a. Navigate to the **Configuration** workspace.
 - b. Click to expand **Monitoring**.
 - c. Select **Custom Poller Configuration**.
 - d. Navigate to the **Custom Poller Collections** tab.
You should see the list of Custom Poller Collections that were created.

Chapter 11: Use Case: SNMP v1 or v2c Management Through Net-SNMP Proxy

NNMi can perform the following operations through one or more Net-SNMP proxies:

- Management of remote nodes using SNMP v1/2c Get, GetNext, and GetBulk requests
- Using Net-SNMP proxy as a protocol translator for Get requests between NNMi and the remote node. This works between SNMP versions 1 and v2c as well as between IPv4 and IPv6.
- Forwarding of SNMP v1/v2c traps from remote nodes to NNMi

All the pieces of SNMP information available to NNMi by directly polling a node are also available to NNMi when polling is done through a Net-SNMP proxy—provided NNMi is configured to have access to the entire MIB tree of the node in the Net-SNMP proxy.

For more information about configuring a Net-SNMP proxy, see the Net-SNMP online documentation.

The Communication Configuration settings in NNMi determine which remote nodes are managed through a given SNMP proxy. SNMP Proxy settings can be configured for specific node settings, specific SNMP agent settings, region settings, and default settings. When setting up a Net-SNMP proxy to monitor devices behind a firewall, it is important to note the following:

- SNMP traffic from the NNMi server must be allowed to reach the Net-SNMP proxy through the firewall
- ICMP traffic from the NNMi server must be allowed to reach all monitored nodes behind the firewall
- If monitoring VMWare nodes behind the firewall, HTTP or HTTPS traffic must be allowed from the NNMi management server to those nodes through the firewall
- If monitoring with NETCONF (for example, to monitor Juniper devices) behind the firewall, SSH traffic must be allowed from the NNMi management server to those devices through the firewall
- You cannot use the Net-SNMP proxy with iSPIs

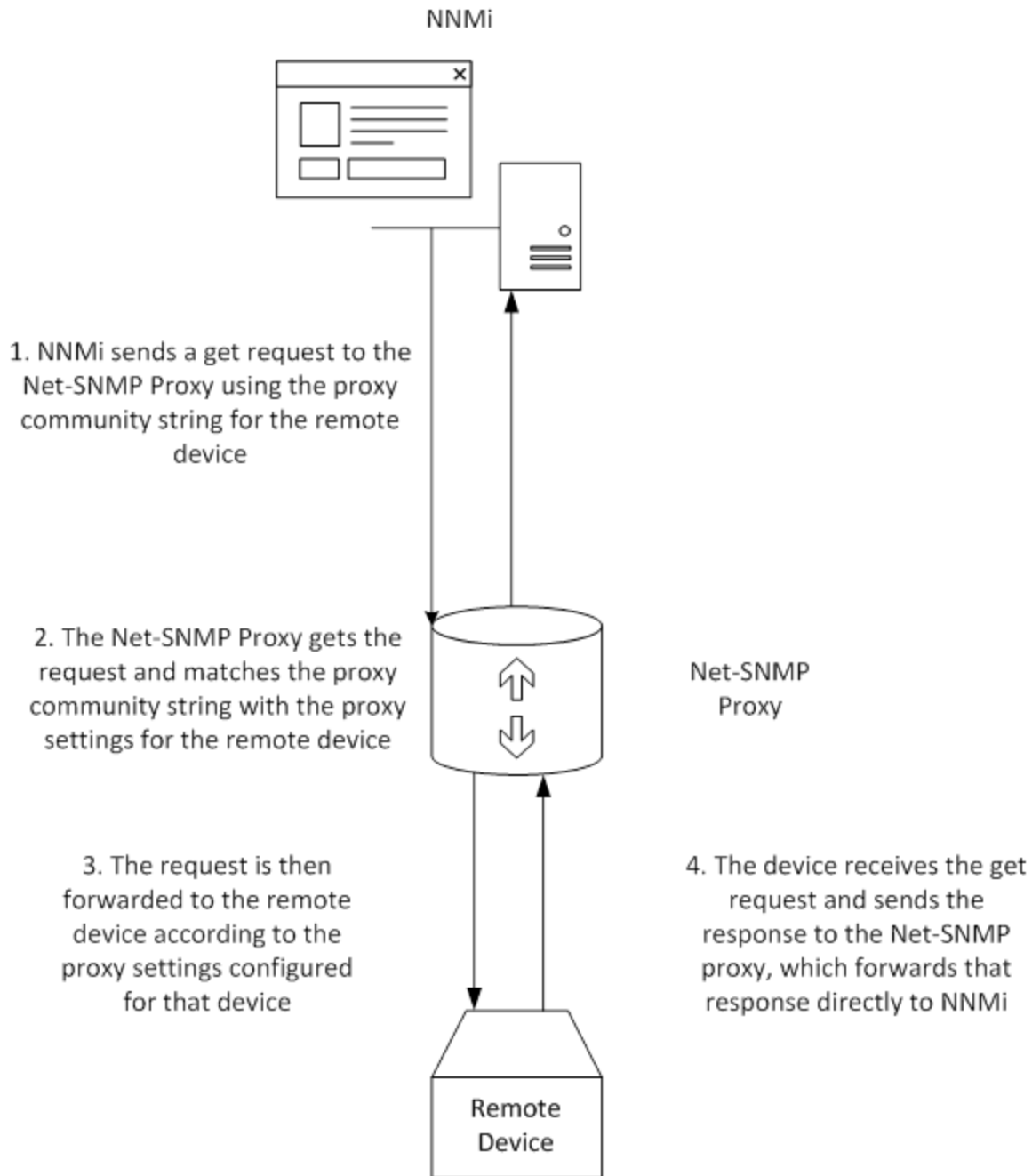
SNMP Get Requests through Net-SNMP Proxy

Net-SNMP handles proxy requests through a community string scheme. For each remote node to be managed, the proxy is configured with a unique community string and a proxy setting that includes the necessary information for the proxy to communicate with the remote device. The NNMi management server and other SNMP clients then use a specific community string to communicate with the associated node through the proxy. In Net-SNMP, this configuration is handled in the `snmpd.conf` file on the Net-SNMP proxy.

In NNMi, the specific community string for a remote node must be configured in the Communication Configuration, Specific Node Settings, or the SNMP Agent settings for a given remote node. Additionally, the Net-SNMP proxy address must be configured as SNMP Proxy Address and the Net-SNMP proxy port must be configured as SNMP Proxy for the remote node. This may be done in any of the following NNMi forms:

- SNMP Agent settings
- Communication Configuration, Specific Node Settings
- Communication Configuration, Region Settings
- Communication Configuration, Default settings

The diagram below illustrates the process of NNMi sending a Get request to a remote device using the Net-SNMP proxy.



Configuring Net-SNMP Proxy Settings Using the Command Line Interface

NNMi supports the addition and modification of SNMP proxy settings through the Command Line Interface (CLI) using the `nnmcommunication.ovpl` command. This feature supports communication configuration

settings for specific nodes, regions and default values. It also supports SNMP agent settings, for use when managing those directly in locked mode. The following example demonstrates how to create, list, update, and delete proxy settings using the CLI.

To create proxy settings for a specific node:

1. Obtain the values for the Net-SNMP proxy IP Address and SNMP Port as well as the proxy community string configured for the remote device. For this example, the following values will be used:
 - Proxy Address: 11.11.11.12
 - Proxy Port: 112
 - Host Name: remotehost.mydomain.com
 - Proxy Community String: cnty_remotehost

2. Open up a terminal window on the NNMi management server and run the following command:

```
nnmcommunication.ovpl createNodeSettings --name remotehost.mydomain.com -
snmpCommunity cnty_remotehost -snmpProxyAddress 11.11.11.12 --snmpProxyPort 112 --
fields name,snmpProxyAddress,snmpProxyPort
```

The command should show the following result:

```
+-----+-----+-----+
| Name                | SNMP Proxy Address | SNMP Proxy Port |
+-----+-----+-----+
| remotehost.mydomain.com | 11.11.11.12      | 112             |
+-----+-----+-----+
```

To list the proxy settings configured for a specific node, run the following command:

```
nnmcommunication.ovpl listNodeSettings --fields name,snmpProxyAddress,snmpProxyPort
```

The command should show a result similar to this:

```
+-----+-----+-----+
| Name                | SNMP Proxy Address | SNMP Proxy Port |
+-----+-----+-----+
| remotehost.mydomain.com | 11.11.11.12      | 112             |
+-----+-----+-----+
```

To update the node-specific settings with proxy address 11.11.11.11 and proxy port 111, run the following command:

```
nnmcommunication.ovpl updateNodeSettings --nodeSetting remotehost.mydomain.com -
snmpProxyAddress 11.11.11.11 --snmpProxyPort 111
```

The command should show a result similar to this:

```
+-----+-----+-----+
| Name                | SNMP Proxy Address | SNMP Proxy Port |
+-----+-----+-----+
```

```
| remotehost.mydomain.com | 11.11.11.11 | 111 |
+-----+-----+-----+
```

To delete the SNMP proxy settings for a node, run the following command.

Note: The `snmpProxyAddress` and `snmpProxyPort` fields are updated to null or empty as no replacement values are provided.

`nmcommunication.ovpl updateNodeSettings --nodeSetting remotehost.mydomain.com --snmpProxyAddress -snmpProxyPort`

The command should show a result similar to this:

```
+-----+-----+-----+
| Name                | SNMP Proxy Address | SNMP Proxy Port |
+-----+-----+-----+
| remotehost.mydomain.com |                    |                  |
+-----+-----+-----+
```

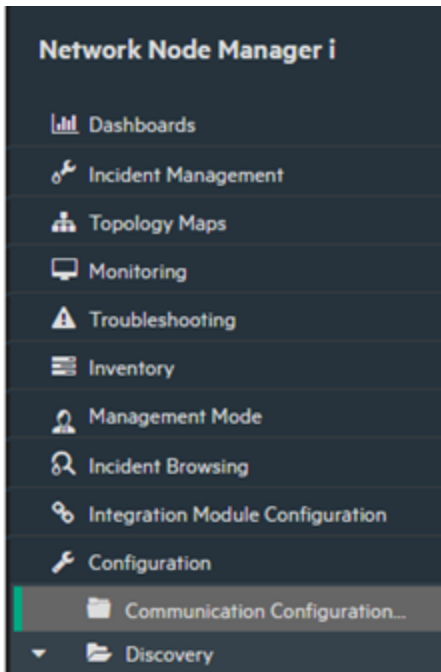
For more information on using the `nmcommunication.ovpl` command to add and modify SNMP Proxy settings, see the reference page for the `nmcommunication.ovpl` command.

Configuring Net-SNMP Proxy Settings Using the Graphical User Interface

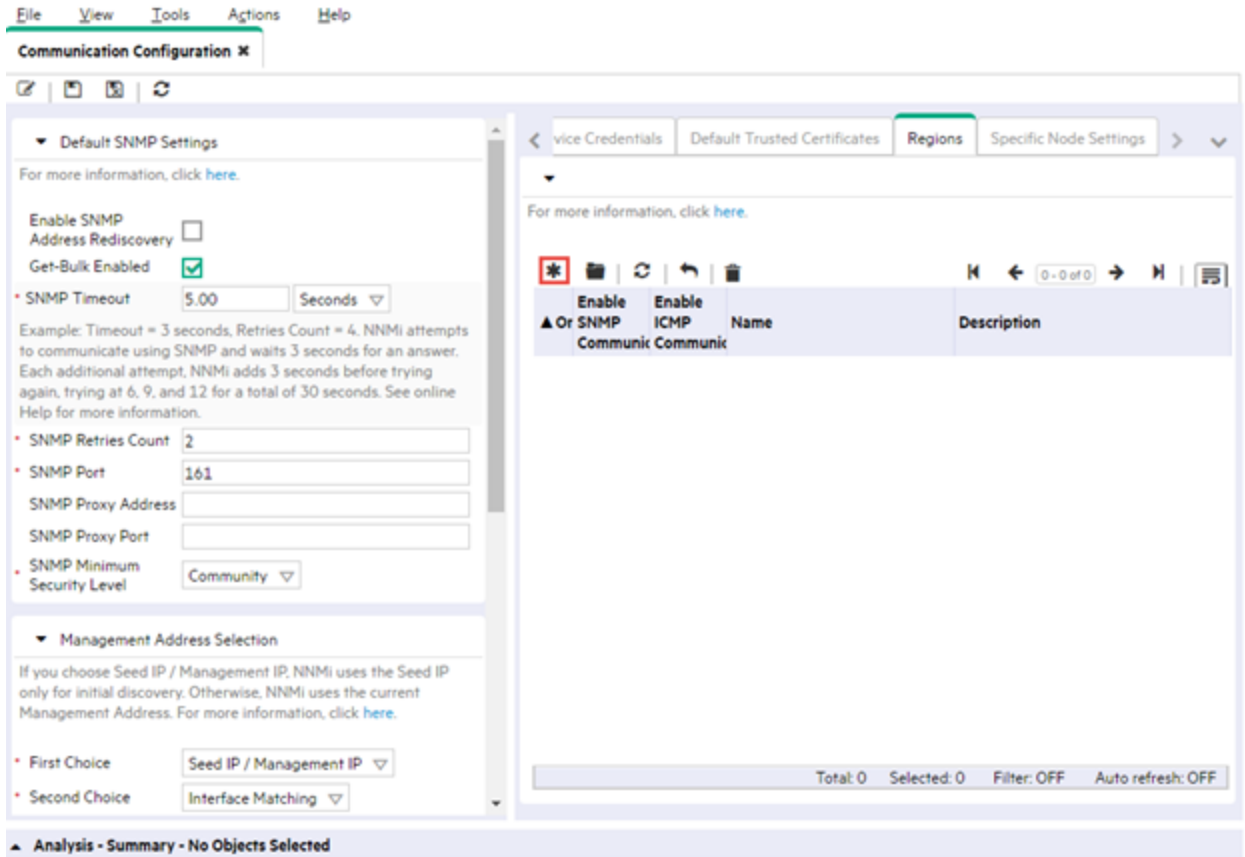
This example will demonstrate how to configure a SNMP proxy settings in a more scalable manner through the NNMi console. The first set of steps will demonstrate how to create a region with the SNMP proxy address and SNMP proxy port, and the second set of steps will demonstrate how to create a specific node setting for a node within the newly created region with the proxy community string. The manual configuration of SNMP proxy settings for the NNMi console is also supported for default settings and specific node settings. When creating a large amount of specific node settings, it is recommended to perform these operations through the `nmcommunication.ovpl` command.

- **Creating the Region with Proxy Settings**

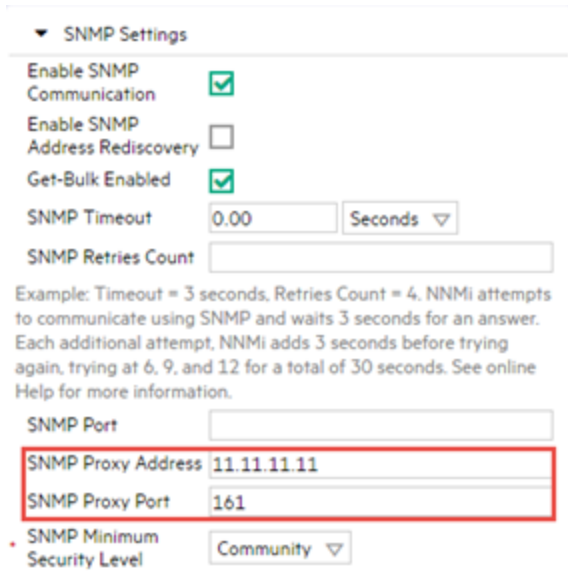
- a. Open Communication Configuration under the Configuration workspace in NNMi:



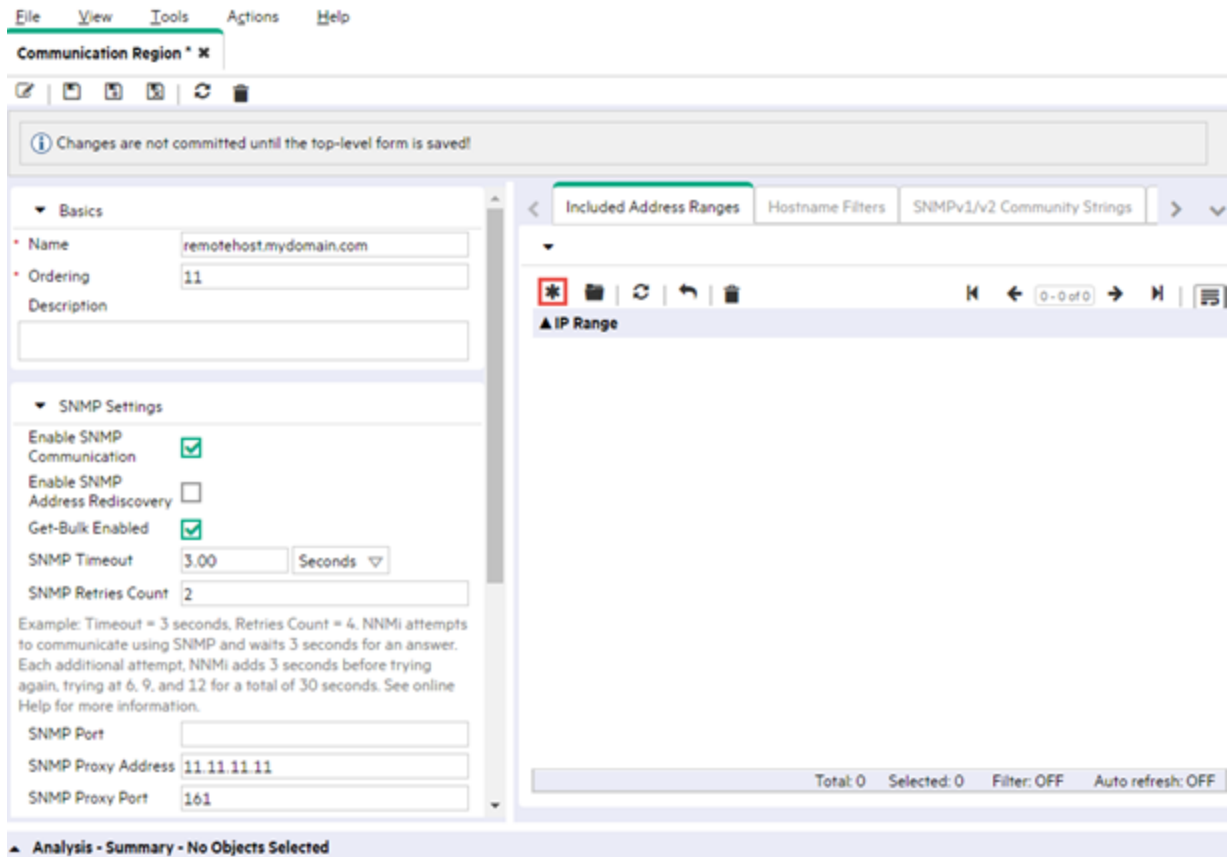
- b. In the Communication Configuration window, navigate to the Regions tab and click on * to add a new communication configuration setting for the remote host.



- c. In the Communication Region window, open the menu for SNMP Settings and type the values of the SNMP Proxy Address and SNMP Proxy port for the region. Fill the other fields as desired.



- d. Complete filling out the other SNMP settings. Go to the Included Address Ranges tab and click * to add an IP address range for the communication region.

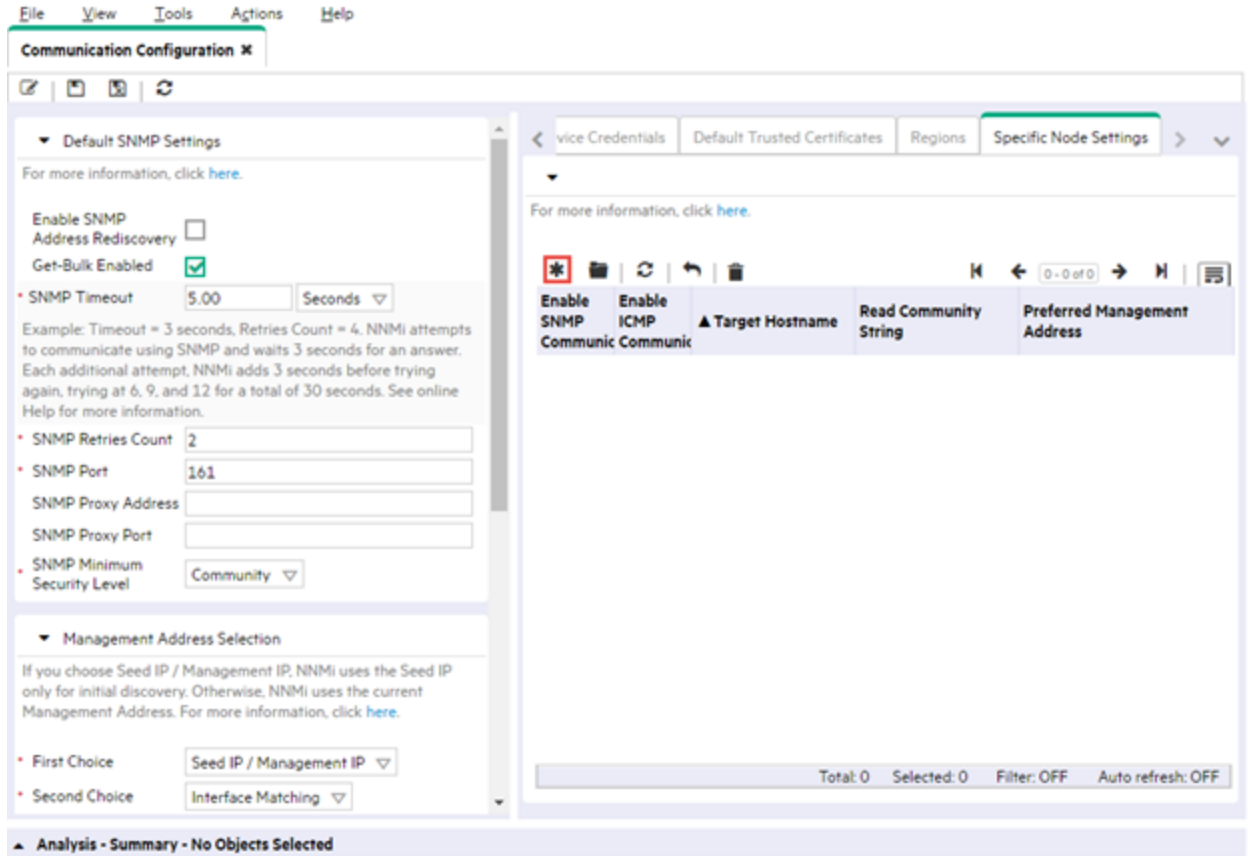


e. Save and close the above forms.

- **Creating the Node Settings for Nodes within the Region**

This example will show how to configure a node setting for a specific node within the new region through the UI. The following instructions are to be repeated for every node in the newly-created communication region. If the number of nodes contained in the region is large, it is recommended that this operation be done through the CLI using the `nnmcommunication.ovp1` tool.

- Re-open the Communication Configuration workspace in NNMi.
- Go to the Specific Node Settings tab in the Communication Configuration window and click * to create a new node setting.



- c. Type the hostname or IP address of the node and the community string configured for the node according to the proxy. Fill in the other communication settings as desired.

Specific Node Settings ✕

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

Basics

Enter the fully-qualified hostname that Spiral Discovery must use in your environment (as registered in your Domain Name System - DNS):

Target Hostname

(Optional) Use if a node has multiple IP addresses:

Preferred Management Address

Description

SNMP Settings

Enable SNMP Communication

Enable SNMP Address Rediscovery

Get-Bulk Enabled

SNMP Timeout Seconds

SNMP Retries Count

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.

SNMPv1/v2 Community Strings | SNMPv3 Settings | Device Credentials | Trusted Certificates

Read Community Strings

Read Community String

Write Community String (Set Community String)

Write Community String

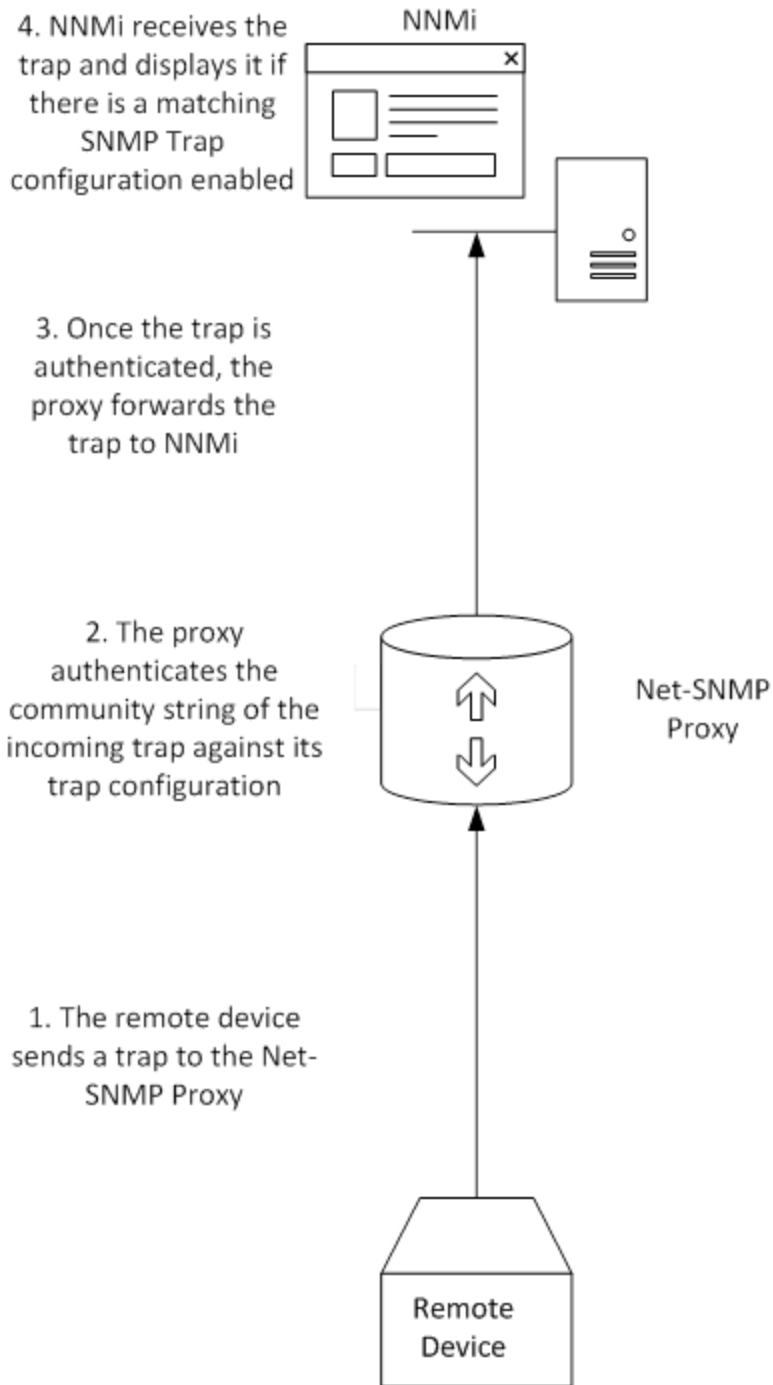
Analysis - Summary - No Objects Selected

- d. Save and close the above forms.

Net-SNMP Trap Forwarding through NNMi

The Net-SNMP proxy follows a community-based scheme for trap forwarding similar to the way it handles Get requests. In order to be able to forward traps to NNMi, the remote device must be configured to send traps to the Net-SNMP proxy. The Net-SNMP proxy must then be configured to receive the traps from the remote device and forward them to NNMi. This is configured in the `snmptrapd.conf` file on the Net-SNMP proxy. Finally, NNMi must have an appropriate SNMP trap configuration enabled for the trap or it will not be displayed in the Incident Browser.

The diagram below shows the process of how traps are forwarded from the remote device to NNMi.



Here are helpful steps to troubleshoot issues with trap forwarding through a Net-SNMP proxy:

- Both the Net-SNMP proxy and the remote nodes must be discovered in NNMi.
- You can send a trap through the Net-SNMP proxy in one of the following two ways:
 - Manually forcing the remote node to produce a trap to forward to the Net-SNMP proxy such as turning an interface off and on again.

- Manually send a trap with the community string of the remote device from a node with a Net-SNMP agent installed to the Net-SNMP proxy using the `snmptrap` command. For example:

```
snmptrap -v 2c -c myCommunity myproxy.mydomain.com 42 coldStart.0
```

If such a node is unavailable, it is possible to run this command on the Net-SNMP proxy and send a trap to itself.

- If the trap is not being received by the NNMi server, check the trap-forwarding configuration of the Net-SNMP proxy and the remote device.
- If the trap is being received by the NNMi server but not being displayed in the UI, verify that there is a SNMP trap configuration enabled in NNMi whose OID matches the OID of the incoming trap.
- It is highly recommended to install a packet analyzing tool such as Wireshark on the NNMi management server and Net-SNMP proxy to verify that SNMP traps are being forwarded from the remote node to NNMi.

Appendix A: Additional Information

This section contains the following appendices:

- "Manually Configuring NNMi for Application Failover" below
- "NNMi Environment Variables" on page 430
- "NNMi and NNM iSPI Default Ports" on page 433
- "NNMi Configuration Issues" on page 467

Manually Configuring NNMi for Application Failover

The steps contained in this appendix provide an alternative to using the NNMi Cluster Setup Wizard to configure application failover.

Note: If you are using application failover with Oracle as your database, you must follow the configuration steps in this appendix, including the following prerequisite action:

You must install the standby server using the "Secondary Server Installation" option. If you installed the standby server as a primary server, uninstall that server and reinstall it using the "Secondary Server Installation" option.

Before uninstalling NNMi, remove any NNMi patches in reverse order, beginning with the most recent patch. The patch removal process varies according to the operating system running on the NNMi management server. See the patch documentation for installation and removal instructions.

To manually configure application failover, perform the following steps:

1. Run `ovstop` on both nodes.
2. 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 following procedure:

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

- a. Edit the following file:
 - *Windows:* `%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.

`com.hp.ov.nms.cluster.name=MyCluster`
- c. Add the hostnames 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`

- d. *Optional.* Define other `com.hp.ov.nms.cluster*` parameters within the `nms-cluster.properties` file. Follow the instructions contained within the `nms-cluster.properties` file for modifying each parameter

Note: If you are using application failover with Oracle as your database, NNMI ignores the database parameters contained in the `nms-cluster.properties` file.

3. Depending on the approach you take, complete the instructions shown in "[Working with Certificates in Application Failover Environments](#)" on page 257.

Caution: 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 from [step 3](#)*

4. 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
```

5. Run the following command on both server X and server Y: **nnmcluster**

Each server should display something similar to the following:

```
===== Current cluster state =====
```

```
State ID: 000000001000000005
```

```
Date/Time: 15 Mar 2011 - 09:37:58 (GMT-0600)
```

```
Cluster name: ThisCluster (key CRC:626,187,650)
```

```
Automatic failover: Enabled
```

```
NNM database type: Embedded
```

```
NNM configured ACTIVE node is: NO_ACTIVE
```

```
NNM current ACTIVE node is: NO_ACTIVE
```

```
Cluster members are:
```

Local?	NodeType	State	OvStatus	Hostname/Address
-----	-----	-----	-----	-----
* REMOTE	ADMIN	n/a	n/a	serverX.xxx.yyy.yourcompany.com/16.78.61.68:7800
(SELF)	ADMIN	n/a	n/a	serverY.xxx.yyy.yourcompany.com/16.78.61.71:7800

The display should list both server X and server Y. If information about both nodes are not displayed, the nodes are not communicating with each other. Here are some things to check for and correct before continuing:

- The Cluster names might be different on server X and server Y.
- The key CRCs might be different on server X and server Y. Check the contents of the following files on both server X and server Y:
Windows: %NnmDataDir%\shared\nnm\conf\nnmcluster\cluster.keystore
Linux: \$NnmDataDir/shared/nnm/conf/nnmcluster/cluster.keystore
- A firewall on server X or server Y might be preventing the nodes from communicating.
- Make sure you merged the `nnm-trust.p12` files. You should see this error displayed after running the `nnmcluster` command.
- Server X and server Y are running different operating systems. For example, suppose server X is running a Linux operating system and server Y is running a Windows operating system. You should see this error displayed after running the `nnmcluster` command.
- Server X and server Y are running different NNMi versions. For example, suppose server X is running 10.30 and server Y is running NNMi 10.30. You should see this error displayed after running the `nnmcluster` command.

6. On server X, start the NNMi cluster manager:

nnmcluster -daemon

Note: 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 more information, see the `nnmcluster` reference page, or the Linux manpage.

7. Wait a few minutes for server X to become the first active node in the cluster. Run the `nnmcluster -display` command on server X and search the displayed results for the term ACTIVE as in ACTIVE_NNM_STARTING or ACTIVE_SomeOtherState. Do not continue with [step 8](#) until you know that server X is the active node.
8. On server Y, start the NNMi cluster manager:

nnmcluster -daemon

Note: 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 more information, see the `nnmcluster` reference page, or the Linux manpage.

9. 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).
10. Instruct network operations center (NOC) personnel to configure their devices to send traps to both server X and server Y. While server X (active) is running, it processes the forwarded traps and server Y (standby) ignores the forwarded traps.

NNMi Environment Variables

HPE Network Node Manager i Software (NNMi) provides many environment variables that are available for your use in navigating the file system and writing scripts.

This appendix contains the following topics:

- ["Environment Variables Used in This Document" below](#)
- ["Other Available Environment Variables" on the next page](#)

Environment Variables Used in This Document

This document primarily uses the following two NNMi environment variables to reference file and directory locations. This list shows the default values. Actual values depend on the selections that you made during NNMi installation.

- *Windows Server*:
 - %NmInstallDir%: <drive>\Program Files (x86)\HP\HP BTO Software
 - %NmDataDir%: <drive>\ProgramData\HP\HP BTO Software

Note: On Windows systems, the NNMi installation process creates these system environment variables, so they are always available to all users.

- *Linux*:
 - \$NmInstallDir: /opt/OV
 - \$NmDataDir: /var/opt/OV

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

Additionally, this document references some of the NNMi environment variables that you can source as part of your user log-on configuration on the NNMi management server. These variables are of the form NNM_*. For information about this extended list of NNMi environment variables, see ["Other Available Environment Variables" below](#).

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 examples:

- Windows: "C:\Program Files (x86)\HP\HP BTO Software\bin\nm.envvars.bat"
- Linux: . /opt/OV/bin/nm.envvars.sh

After you run the command for your operating system, you can use the NNMi environment variables shown in [Environment Variable Default Locations for the Windows Operating System](#) or [Environment Variable Default Locations for Linux Operating Systems](#) to get to commonly used NNMi file locations.

Environment Variable Default Locations for the Windows Operating System

Variable	Windows (example)
%NNM_BIN%	C:\Program Files (x86)\HP\HP BTO Software\bin
%NNM_CONF%	C:\ProgramData\HP\HP BTO Software\conf
%NNM_DATA%	C:\ProgramData\HP\HP BTO Software\
%NNM_DB%	C:\ProgramData\HP\HP BTO Software\shared\nm\databases
%NNM_JAVA%	It is set to the java.exe location of the JDK installed on the NNMi management server
%NNM_JAVA_PATH_SEP%	;

Environment Variable Default Locations for the Windows Operating System, continued

Variable	Windows (example)
%NNM_JRE%	C:\Program Files (x86)\HP\HP BTO Software\nonOV\jdk\hpsw
%NNM_LOG%	C:\ProgramData\HP\HP BTO Software\log
%NNM_LRF%	C:\ProgramData\HP\HP BTO Software\shared\nnm\lrf
%NNM_PRIV_LOG%	C:\ProgramData\HP\HP BTO Software\log
%NNM_PROPS%	C:\ProgramData\HP\HP BTO Software\shared\nnm\conf\props
%NNM_SHARED_CONF%	C:\ProgramData\HP\HP BTO Software\shared\nnm\conf
%NNM_SHARE_LOG%	C:\ProgramData\HP\HP BTO Software\log
%NNM_SNMP_MIBS%	C:\Program Files (x86)\HP\HP BTO Software\misc\nnm\snmp-mibs
%NNM_TMP%	C:\ProgramData\HP\HP BTO Software\tmp
%NNM_USER_SNMP_MIBS%	C:\ProgramData\HP\HP BTO Software\shared\nnm\user-snmp-mibs
%NNM_WWW%	C:\ProgramData\HP\HP BTO Software\shared\nnm\www

Environment Variable Default Locations for Linux Operating Systems

Variable	Linux
\$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	It is set to the java.bin location of the JDK installed on the NNMI management server
\$NNM_JAVA_PATH_SEP	:
\$NNM_JRE	/opt/OV/nonOV/jdk/nnm
\$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

Environment Variable Default Locations for Linux Operating Systems, continued

Variable	Linux
\$NNM_SNMP_MIBS	/opt/OV/misc/nnm/snmp-mibs
\$NNM_TMP	/var/opt/OV/tmp
\$NNM_USER_SNMP_MIBS	/var/opt/OV/shared/nnm/user-snmp-mibs
\$NNM_WWW	/var/opt/OV/shared/nnm/www

NNMi and NNM iSPI Default Ports

This appendix shows the default ports that NNMi and the NNM iSPIs use in network communications. If port conflicts occur between products, you can change most of these port numbers as shown in the *Change Configuration* column.

In addition, subsequent topics document the ports used by the individual HPE Network Management Software products.

- ["HPE Network Node Manager i Software Ports" on page 434](#)
- ["NNM iSPI for MPLS Ports" on page 445](#)
- ["NNM iSPI for IP Telephony Ports" on page 448](#)
- ["NNM iSPI for IP Multicast Ports" on page 451](#)
- ["NNM iSPI Performance for Traffic Ports" on page 454](#)
- ["NNM iSPI Performance for QA Ports" on page 461](#)
- ["NNM iSPI Performance for Metrics and NPS Ports" on page 465](#)
- ["NNM iSPI NET Ports" on page 466](#)

HPE Network Node Manager i Software Ports

The NNMi ports fit into the following categories:

- [Ports Used on the NNMi Management Server](#)
- [Ports Used for Communication Between the NNMi Management Server and Other Systems](#)
- [Required Accessible Sockets for Global Network Management](#)

Ports Used on the NNMi Management Server

The following table shows 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. See the *nmm.ports* reference page, or the Linux manpage, for more information.

Note: For application failover to work successfully, open TCP ports 7800-7810. For the application failover feature to function correctly, the active and standby NNMi management servers must have unrestricted network access to each other.

Ports Used on the NNMi Management Server

Port	Type	Name	Purpose	Change Configuration
80	TCP	nmsas.server.port.web.http	Default HTTP port - used for Web UI & Web Services - In GNM configurations NNMi uses this port to establish communicatio	Modify the %NNM_CONF%\nmm\props\nms-local.properties file (Windows) or \$NNM_CONF/nmm/props/nms-local.properties file (Linux). You can also change this during installation.

Ports Used on the NNMi Management Server, continued

Port	Type	Name	Purpose	Change Configuration
			<p>n from the global manager to the regional manager</p> <p>- Once this port is open, it becomes bi-directional</p>	
162	UDP	trapPort	SNMP trap port	Modify using the <code>nmtrapconfig.ovpl</code> Perl script. See the <i>nmtrapconfig.ovpl</i> reference page, or the Linux manpage, for more information.
443	TCP	nmsas.server.port.web.https	Default secure HTTPS port (SSL) - used for Web UI & Web Services	Modify the <code>%NNM_CONF%\nmm\props\nms-local.properties</code> file (Windows) or <code>\$(NNM_CONF)/nmm/props/nms-local.properties</code> file (Linux).
1098	TCP	nmsas.server.port.naming.rmi	<p>- Used by NNMi command line tools to communicate with a variety of services used by NNMi</p> <p>- HPE recommends</p>	Modify the <code>%NNM_CONF%\nmm\props\nms-local.properties</code> file (Windows) or <code>\$(NNM_CONF)/nmm/props/nms-local.properties</code> file (Linux).

Ports Used on the NNMi Management Server, continued

Port	Type	Name	Purpose	Change Configuration
			configuring the system firewall to restrict access to these ports to localhost only	
1099	TCP	nmsas.server.port.naming.port	<p>- Used by NNMi command line tools to communicate with a variety of services used by NNMi</p> <p>- HPE recommends configuring the system firewall to restrict access to these ports to localhost only</p>	Modify the %NNM_CONF%\nsm\props\nms-local.properties file (Windows) or \$NNM_CONF/nsm/props/nms-local.properties file (Linux).
3873	TCP	nmsas.server.port.remoting.ejb3	- Used by NNMi command line tools to	Modify the %NNM_CONF%\nsm\props\nms-local.properties file (Windows) or \$NNM_CONF/nsm/props/nms-local.properties file (Linux).

Ports Used on the NNMi Management Server, continued

Port	Type	Name	Purpose	Change Configuration
			<p>communicate with a variety of services used by NNMi</p> <p>- HPE recommends configuring the system firewall to restrict access to these ports to localhost only</p>	
4444	TCP	nmsas.server.port.jmx.jrmp	<p>- Used by NNMi command line tools to communicate with a variety of services used by NNMi</p> <p>- HPE recommends configuring the system firewall to restrict access to</p>	<p>Modify the %NNM_CONF%\nrm\props\nms-local.properties file (Windows) or \$NNM_CONF/nrm/props/nms-local.properties file (Linux).</p>

Ports Used on the NNMi Management Server, continued

Port	Type	Name	Purpose	Change Configuration
			these ports to localhost only	
4445	TCP	nmsas.server.port.jmx.rmi	<p>- Used by NNMi command line tools to communicate with a variety of services used by NNMi</p> <p>- HPE recommends configuring the system firewall to restrict access to these ports to localhost only</p>	Modify the %NNM_CONF%\nsm\props\nms-local.properties file (Windows) or \$NNM_CONF/nsm/props/nms-local.properties file (Linux).
4446	TCP	nmsas.server.port.invoker.unified	<p>- Used by NNMi command line tools to communicate with a variety of services used by NNMi</p>	Modify the %NNM_CONF%\nsm\props\nms-local.properties file (Windows) or \$NNM_CONF/nsm/props/nms-local.properties file (Linux).

Ports Used on the NNMi Management Server, continued

Port	Type	Name	Purpose	Change Configuration
			- HPE recommends configuring the system firewall to restrict access to these ports to localhost only	
4457	TCP	nmsas.server.port.hq	<p>- Used for un-encrypted Global Network Management traffic.</p> <p>- Messaging travels from the global manager to the regional manager</p> <p>- Once this port is open, it becomes bi-directional</p>	Modify the %NNM_CONF%\nmm\props\nms-local.properties file (Windows) or \$NNM_CONF/nmm/props/nms-local.properties file (Linux).
4459	TCP	nmsas.server.port.hq.ssl	- Used for encrypted Global	Modify the %NNM_CONF%\nmm\props\nms-local.properties file (Windows) or \$NNM_CONF/nmm/props/nms-local.properties file (Linux).

Ports Used on the NNMi Management Server, continued

Port	Type	Name	Purpose	Change Configuration
			<p>Network Management traffic.</p> <ul style="list-style-type: none"> - Messaging travels from the global manager to the regional manager - Once this port is open, it becomes bi-directional 	
4712	TCP	nmsas.server.port.ts.recovery	Internal transaction service port	Modify the %NNM_CONF%\nsm\props\nms-local.properties file (Windows) or \$NNM_CONF/nsm/props/nms-local.properties file (Linux).
4713	TCP	nmsas.server.port.ts.status	Internal transaction service port	Modify the %NNM_CONF%\nsm\props\nms-local.properties file (Windows) or \$NNM_CONF/nsm/props/nms-local.properties file (Linux).
4714	TCP	nmsas.server.port.ts.id	Internal transaction service port	Modify the %NNM_CONF%\nsm\props\nms-local.properties file (Windows) or \$NNM_CONF/nsm/props/nms-local.properties file (Linux).
5432	TCP	com.hp.ov.nms.postgres.port	This PostgreSQL port is the port the embedded	Modify the %NNM_CONF%\nsm\props\nms-local.properties file (Windows) or \$NNM_CONF/nsm/props/nms-local.properties file (Linux).

Ports Used on the NNMi Management Server, continued

Port	Type	Name	Purpose	Change Configuration
			database listens on for this NNMi management server.	
7800-7810	TCP		<ul style="list-style-type: none"> - JGroups ports for application failover - If application failover is not used, HPE recommends configuring the system firewall to restrict access to these ports 	Modify the %NNM_CONF%\nsm\props\nms-cluster.properties file (Windows) or \$NNM_CONF/nsm/props/nms-cluster.properties file (Linux).
8886	TCP	OVsPMD_MGMT	NNMi ovspmd (process manager) management port	Modify the /etc/services file.
8887	TCP	OVsPMD_REQ	NNMi ovsmpr (process manager) request port	Modify the /etc/services file.

Ports Used on the NNMi Management Server, continued

Port	Type	Name	Purpose	Change Configuration
8989	TCP	com.hp.ov.nms.events.action.server.port	Action server port	Modify the %NmInstallDir%\misc\nnm\props\shared\nnmaction.properties file (Windows) or \$NmInstallDir/misc/nnm/props/shared/nnmaction.properties file (Linux).

Ports Used for Communication Between the NNMi Management Server and Other Systems

The following table shows some of the ports NNMi uses to communicate with other systems. If a firewall separates NNMi from these systems, you must open many of these ports in the firewall. The actual set of ports depends on the set of integrations you configured to use with NNMi and how you configured those integrations. If column 4 indicates *Client*, NNMi connects or sends to this port; if column 4 indicates *Server*, NNMi listens on this port.

Ports Used for Communication Between the NNMi Management Server and Other Systems

Port	Type	Purpose	Client, Server
80	TCP	Default HTTP port for NNMi; used for Web UI and Web Services	Server
80	TCP	Default HTTP port for NNMi connecting to other applications. The actual port depends on NNMi configuration.	Client
161	UDP	SNMP request port	Client
162	UDP	SNMP trap port - traps received by NNMi	Server
162	UDP	SNMP trap port; Trap Forwarding, Northbound Interface, or NetCool integrations	Client
389	TCP	Default LDAP port	Client
395	UDP	nGenius Probe SNMP trap port	Client

Ports Used for Communication Between the NNMi Management Server and Other Systems, continued

Port	Type	Purpose	Client, Server
443	TCP	Default secure HTTPS port for NNMi connecting to other applications; the actual port depends on NNMi configuration. Default HTTPS port for HPOM on Windows	Client
443	TCP	Default secure HTTPS port; used for Web UI and Web Services	Server
636	TCP	Default secure LDAP port (SSL)	Client
1741	TCP	Default CiscoWorks LMS web services port	Client
4457	TCP	Used for un-encrypted Global Network Management traffic. The connection is from the global manager to the regional manager.	Client, Server
4459	TCP	Used for encrypted Global Network Management traffic. The connection is from the global manager to the regional manager.	Client, Server
7800-7810	TCP	JGroups ports for application failover	Client and Server
8004	TCP	Default HTTP port for NNMi if another web server already has port 80. Used for Web UI and Web Services. Verify the actual HTTP port for your NNMi management server.	Server
8080	TCP	Default HTTP port for connecting to NA if installed on the same system as NNMi. Default HTTPS port for UCMDB web services	Client
8443 or 8444	TCP	Default HTTP port for connecting to HPOM for UNIX	Client
9300	TCP	Default HTTP port for connecting to NNM iSPI Performance for Metrics	Client
50000	TCP	Default HTTPS port for connecting to SIM	Client

Note: If you configure NNMi to use ICMP fault polling or ping sweep for discovery, configure the firewall to pass ICMP packets through the firewall.

Note: The Web Services approach for the NNMi-HPOM integration does not work through a firewall, however the NNMi-HPOM integration using the Northbound Interface does work through a firewall.

Required Accessible Sockets for Global Network Management

The following table shows the well-known ports that need to be accessible from a global NNMi management server to a regional 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.

Required Accessible Sockets for Global Network Management

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

NNM iSPI for MPLS Ports

The following table shows the ports the HPE Network Node Manager iSPI for MPLS Software uses on the management server. In case of port conflicts, almost all of these port numbers can be changed using the `server.properties` file located at:

`%NnmDataDir%/nmsas/mpls/server.properties`.

Ports Used on the HPE Network Node Manager iSPI for MPLS Software Management Server

Port	Type	Name	Purpose	Change Configuration
5432	TCP	<code>com.hp.ov.nms.postgres.port</code>	This PostgreSQL port is the port the embedded database listens on for this NNMi management server. The port is expected to be the same as that configured for NNMi in the <code>nms-local.properties</code> file.	N/A
24040	TCP	<code>nmsas.server.port.web.http</code>	Default HTTP port - used for Web UI.	Modify the <code>%NnmDataDir%\nmsas\mpls\server.properties</code> file (Windows) or <code>\$NnmDataDir/nmsas/mpls/server.properties</code> file (Linux). You can also change this during installation.
24041	TCP	<code>nmsas.server.port.remoting.ejb3</code>	Default EJB3 remoting connector port	Modify the <code>%NnmDataDir%\nmsas\mpls\server.properties</code> file (Windows) or <code>\$NnmDataDir/nmsas/mpls/server.properties</code> file (Linux).
24043	TCP	<code>nmsas.server.port.web.https</code>	Default secure HTTPS port (SSL) - used for Web UI.	Modify the <code>%NnmDataDir%\nmsas\mpls\server.properties</code> file (Windows) or <code>\$NnmDataDir/nmsas/mpls/server.properties</code> file (Linux). You can also change this during

Ports Used on the HPE Network Node Manager iSPI for MPLS Software Management Server, continued

Port	Type	Name	Purpose	Change Configuration
				installation.
24044	TCP	nmsas.server.port.jmx.jmp	Default RMI object port (JRMP invoker)	Modify the %NnmDataDir%\nmsas\mpls\server.properties file (Windows) or \$NnmDataDir/nmsas/mpls/server.properties file (Linux).
24045	TCP	nmsas.server.port.invoker.unified	Default RMI remoting server connector port	Modify the %NnmDataDir%\nmsas\mpls\server.properties file (Windows) or \$NnmDataDir/nmsas/mpls/server.properties file (Linux).
24046	TCP	nmsas.server.port.naming.port	Default bootstrap JNP service port (JNDI provider)	Modify the %NnmDataDir%\nmsas\mpls\server.properties file (Windows) or \$NnmDataDir/nmsas/mpls/server.properties file (Linux). You can also change this during installation.
24047	TCP	nmsas.server.port.hq	Used for un-encrypted Global Network Management traffic.	Modify the %NnmDataDir%\nmsas\mpls\server.properties file (Windows) or \$NnmDataDir/nmsas/mpls/server.properties file (Linux).
24048	TCP	nmsas.server.port.jmx.rmi	Default RMI pooled invoker port	Modify the %NnmDataDir%\nmsas\mpls\server.properties file (Windows) or \$NnmDataDir/nmsas/mpls/server.properties file (Linux).

Ports Used on the HPE Network Node Manager iSPI for MPLS Software Management Server, continued

Port	Type	Name	Purpose	Change Configuration
24049	TCP	nmsas.server.port.naming.rmi	Default port for RMI naming service	Modify the %NnmDataDir%\nmsas\mpls\server.properties file (Windows) or \$NnmDataDir/nmsas/mpls/server.properties file (Linux).
24092	TCP	nmsas.server.port.hq.ssl	Used for encrypted Global Network Management traffic.	Modify the %NnmDataDir%\nmsas\mpls\server.properties file (Windows) or \$NnmDataDir/nmsas/mpls/server.properties file (Linux).
24712	TCP	nmsas.server.port.ts.recovery	Default recovery port used by the Transaction service.	Modify the %NnmDataDir%\nmsas\mpls\server.properties file (Windows) or \$NnmDataDir/nmsas/mpls/server.properties file (Linux).
24713	TCP	nmsas.server.port.ts.status	Default status port used by the Transaction service.	Modify the %NnmDataDir%\nmsas\mpls\server.properties file (Windows) or \$NnmDataDir/nmsas/mpls/server.properties file (Linux).
24714	TCP	nmsas.server.port.ts.id	Default port used by the Transaction service.	Modify the %NnmDataDir%\nmsas\mpls\server.properties file (Windows) or \$NnmDataDir/nmsas/mpls/server.properties file (Linux).

NNM iSPI for IP Telephony Ports

The following table shows the ports the NNM iSPI for IP Telephony uses on the management server. In case of port conflicts, almost all of these port numbers can be changed using the `server.properties` file located at: `%NnmDataDir%/nmsas/ipt/server.properties`.

Ports Used on the NNM iSPI for IP Telephony Management Server

Port	Type	Name	Purpose	Change Configuration
5432	TCP	<code>com.hp.ov.nms.postgres.port</code>	This PostgreSQL port is the port the embedded database listens on for this NNMi management server. The port is expected to be the same as that configured for NNMi in the <code>nms-local.properties</code> file.	N/A
10080	TCP	<code>nmsas.server.port.web.http</code>	Default HTTP port - used for Web UI.	Modify the <code>%NnmDataDir%\nmsas\ipt\server.properties</code> file (Windows) or <code>\$NnmDataDir/nmsas/ipt/server.properties</code> file (Linux). You can also change this during installation.
10083	TCP	<code>nmsas.server.port.naming.rmi</code>	Default port for RMI naming service	Modify the <code>%NnmDataDir%\nmsas\ipt\server.properties</code> file (Windows) or <code>\$NnmDataDir/nmsas/ipt/server.properties</code> file (Linux).
10084	TCP	<code>nmsas.server.port.jmx.jrmp</code>	Default RMI object port (JRMP invoker)	Modify the <code>%NnmDataDir%\nmsas\ipt\server.properties</code> file (Windows) or <code>\$NnmDataDir/nmsas/ipt/server.properties</code> file (Linux).

Ports Used on the NNM iSPI for IP Telephony Management Server, continued

Port	Type	Name	Purpose	Change Configuration
10085	TCP	nmsas.server.port.jmx.rmi	Default RMI pooled invoker port	Modify the %NnmDataDir%\nmsas\ipt\server.properties file (Windows) or \$NnmDataDir/nmsas/ipt/server.properties file (Linux).
10086	TCP	nmsas.server.port.invoker.unified	Default RMI remoting server connector port	Modify the %NnmDataDir%\nmsas\ipt\server.properties file (Windows) or \$NnmDataDir/nmsas/ipt/server.properties file (Linux).
10087	TCP	nmsas.server.port.hq	Used for un-encrypted Global Network Management traffic.	Modify the %NnmDataDir%\nmsas\ipt\server.properties file (Windows) or \$NnmDataDir/nmsas/ipt/server.properties file (Linux).
10089	TCP	nmsas.server.port.remoting.ejb3	Default EJB3 remoting connector port	Modify the %NnmDataDir%\nmsas\ipt\server.properties file (Windows) or \$NnmDataDir/nmsas/ipt/server.properties file (Linux).
10092	TCP	nmsas.server.port.hq.ssl	Used for encrypted Global Network Management traffic.	Modify the %NnmDataDir%\nmsas\ipt\server.properties file (Windows) or \$NnmDataDir/nmsas/ipt/server.properties file (Linux).
10099	TCP	nmsas.server.port.naming.port	Default bootstrap JNP service port (JNDI provider)	Modify the %NnmDataDir%\nmsas\ipt\server.properties

Ports Used on the NNM iSPI for IP Telephony Management Server, continued

Port	Type	Name	Purpose	Change Configuration
				file (Windows) or \$NnmDataDir/nmsas/ipt/server.properties file (Linux). You can also change this during installation.
10443	TCP	nmsas.server.port.web.https	Default secure HTTPS port (SSL) - used for Web UI.	Modify the %NnmDataDir%\nmsas\ipt\server.properties file (Windows) or \$NnmDataDir/nmsas/ipt/server.properties file (Linux). You can also change this during installation.
14712	TCP	nmsas.server.port.ts.recovery	Default recovery port used by the Transaction service.	Modify the %NnmDataDir%\nmsas\ipt\server.properties file (Windows) or \$NnmDataDir/nmsas/ipt/server.properties file (Linux).
14713	TCP	nmsas.server.port.ts.status	Default status port used by the Transaction service.	Modify the %NnmDataDir%\nmsas\ipt\server.properties file (Windows) or \$NnmDataDir/nmsas/ipt/server.properties file (Linux).
14714	TCP	nmsas.server.port.ts.id	Default port used by the Transaction service.	Modify the %NnmDataDir%\nmsas\ipt\server.properties file (Windows) or \$NnmDataDir/nmsas/ipt/server.properties file (Linux).

NNM iSPI for IP Multicast Ports

The following table shows the ports the NNM iSPI for IP Multicast uses on the management server. In case of port conflicts, almost all of these port numbers can be changed using the `server.properties` file located at: `%NnmDataDir%/nmsas/multicast/server.properties`.

Ports Used on the NNM iSPI for IP Multicast Management Server

Port	Type	Name	Purpose	Change Configuration
5432	TCP	<code>com.hp.ov.nms.postgres.port</code>	This PostgreSQL port is the port the embedded database listens on for this NNMi management server. The port is expected to be the same as that configured for NNMi in the <code>nms-local.properties</code> file.	N/A
8084	TCP	<code>nmsas.server.port.web.http</code>	Default HTTP port - used for Web UI.	Modify the <code>%NnmDataDir%\nmsas\multicast\server.properties</code> file (Windows) or <code>\$NnmDataDir/nmsas/multicast/server.properties</code> file (Linux). You can also change this during installation.
14083	TCP	<code>nmsas.server.port.naming.rmi</code>	Default port for RMI naming service	Modify the <code>%NnmDataDir%\nmsas\multicast\server.properties</code> file (Windows) or <code>\$NnmDataDir/nmsas/multicast/server.properties</code> file (Linux).
14084	TCP	<code>nmsas.server.port.jmx.jrmp</code>	Default RMI object port (JRMP invoker)	Modify the <code>%NnmDataDir%\nmsas\multicast\server.properties</code> file (Windows) or <code>\$NnmDataDir/nmsas/multicast/server.properties</code> file (Linux).

Ports Used on the NNM iSPI for IP Multicast Management Server, continued

Port	Type	Name	Purpose	Change Configuration
14085	TCP	nmsas.server.port.jmx.rmi	Default RMI pooled invoker port	Modify the %NnmDataDir%\nmsas\multicast\server.properties file (Windows) or \$NnmDataDir/nmsas/multicast/server.properties file (Linux).
14086	TCP	nmsas.server.port.invoker.unified	Default RMI remoting server connector port	Modify the %NnmDataDir%\nmsas\multicast\server.properties file (Windows) or \$NnmDataDir/nmsas/multicast/server.properties file (Linux).
14087	TCP	nmsas.server.port.hq	Used for un-encrypted Global Network Management traffic.	Modify the %NnmDataDir%\nmsas\multicast\server.propertiesfile (Windows) or \$NnmDataDir/nmsas/multicast/server.properties file (Linux).
14089	TCP	nmsas.server.port.remoting.ejb3	Default EJB3 remoting connector port	Modify the %NnmDataDir%\nmsas\multicast\server.properties file (Windows) or \$NnmDataDir/nmsas/multicast/server.properties file (Linux).
14092	TCP	nmsas.server.port.hq.ssl	Used for encrypted Global Network Management traffic.	Modify the %NnmDataDir%\nmsas\multicast\server.propertiesfile (Windows) or \$NnmDataDir/nmsas/multicast/server.properties file (Linux).
14099	TCP	nmsas.server.port.naming.port	Default bootstrap JNP service port (JNDI provider)	Modify the %NnmDataDir%\nmsas\multicast\server.properties

Ports Used on the NNM iSPI for IP Multicast Management Server, continued

Port	Type	Name	Purpose	Change Configuration
				file (Windows) or \$NnmDataDir/nmsas/multicast/server.properties file (Linux). You can also change this during installation.
14102	TCP	nmsas.server.port.ts.id	Default port used by the Transaction service.	Modify the %NnmDataDir%\nmsas\multicast\server.properties file (Windows) or \$NnmDataDir/nmsas/multicast/server.properties file (Linux).
14103	TCP	nmsas.server.port.ts.recovery	Default recovery port used by the Transaction service.	Modify the %NnmDataDir%\nmsas\multicast\server.properties file (Windows) or \$NnmDataDir/nmsas/multicast/server.properties file (Linux).
14104	TCP	nmsas.server.port.ts.status	Default status port used by the Transaction service.	Modify the %NnmDataDir%\nmsas\multicast\server.properties file (Windows) or \$NnmDataDir/nmsas/multicast/server.properties file (Linux).
14443	TCP	nmsas.server.port.web.https	Default secure HTTPS port (SSL) - used for Web UI.	Modify the %NnmDataDir%\nmsas\multicast\server.properties file (Windows) or \$NnmDataDir/nmsas/multicast/server.properties file (Linux). You can also change this during installation.

NNM iSPI Performance for Traffic Ports

The NNM iSPI Performance for Traffic ports fit into the following categories:

- [Ports Used on the NNM iSPI Performance for Traffic Management Server \(Traffic Master\)](#)
- [Ports Used on the NNM iSPI Performance for Traffic Management Server \(Traffic Leaf\)](#)
- [Ports Used for Communication Between the NNM iSPI Performance for Traffic Management Server and Other Systems](#)

Ports Used on the NNM iSPI Performance for Traffic Management Server (Traffic Master)

The following table shows the ports the NNM iSPI Performance for Traffic (Traffic Master component) uses on the management server. In case of port conflicts, almost all of these port numbers can be changed using the `server.properties` file located at: `%NmDataDir%\nmsas/traffic-master/server.properties`.

Ports Used on the NNM iSPI Performance for Traffic Management Server (Traffic Master)

Port	Type	Name	Purpose	Change Configuration
5432	TCP	<code>com.hp.ov.nms.postgres.port</code>	This PostgreSQL port is the port the embedded database listens on for this NNMi management server. The port is expected to be the same as that configured for NNMi in the <code>nms-local.properties</code> file	N/A
12080	TCP	<code>nmsas.server.port.web.http</code>	Default HTTP port - used for Web UI.	Modify the <code>%NmDataDir%\nmsas\traffic-master\server.propertiesfile</code> (Windows) or <code>\$NmDataDir/nmsas/traffic-master/server.properties</code> file (Linux). You can also change this during installation.
12043	TCP	<code>nmsas.server.port.web.https</code>	Default secure HTTPS port (SSL) - used for Web UI.	Modify the <code>%NmDataDir%\nmsas\traffic-master\server.propertiesfile</code> (Windows) or <code>\$NmDataDir/nmsas/traffic-</code>

Ports Used on the NNM iSPI Performance for Traffic Management Server (Traffic Master), continued

Port	Type	Name	Purpose	Change Configuration
				master/server.properties file (Linux). You can also change this during installation.
12083	TCP	nmsas.server.port.naming.rmi	Default port for RMI naming service	Modify the %NnmDataDir%\nmsas\traffic-master\server.propertiesfile (Windows) or \$NnmDataDir/nmsas/traffic-master/server.properties file (Linux).
12084	TCP	nmsas.server.port.jmx.jmp	Default RMI object port (JRMP invoker)	Modify the %NnmDataDir%\nmsas\traffic-master\server.propertiesfile (Windows) or \$NnmDataDir/nmsas/traffic-master/server.properties file (Linux).
12085	TCP	nmsas.server.port.jmx.rmi	Default RMI pooled invoker port	Modify the %NnmDataDir%\nmsas\traffic-master\server.propertiesfile (Windows) or \$NnmDataDir/nmsas/traffic-master/server.properties file (Linux).
12086	TCP	nmsas.server.port.invoker.unified	Default RMI remoting server connector port	Modify the %NnmDataDir%\nmsas\traffic-master\server.propertiesfile (Windows) or \$NnmDataDir/nmsas/traffic-master/server.properties file (Linux).
12087	TCP	nmsas.server.port.hq	Used for un-encrypted Global Network Management traffic.	Modify the %NnmDataDir%\nmsas\traffic-master\server.propertiesfile (Windows) or \$NnmDataDir/nmsas/traffic-master/server.properties file (Linux).
12089	TCP	nmsas.server.port.remoting.ejb3	Default EJB3	Modify the %NnmDataDir%\nmsas\traffic-master\server.propertiesfile (Windows) or \$NnmDataDir/nmsas/traffic-master/server.properties file (Linux).

Ports Used on the NNM iSPI Performance for Traffic Management Server (Traffic Master), continued

Port	Type	Name	Purpose	Change Configuration
12092	TCP	nmsas.server.port.hq.ssl	Used for encrypted Global Network Management traffic.	Modify the %NnmDataDir%\nmsas\traffic-master\server.propertiesfile (Windows) or \$NnmDataDir/nmsas/traffic-master/server.properties file (Linux).
12099	TCP	nmsas.server.port.naming.port	Default bootstrap JNP service port (JNDI provider)	Modify the %NnmDataDir%\nmsas\traffic-master\server.propertiesfile (Windows) or \$NnmDataDir/nmsas/traffic-master/server.properties file (Linux). You can also change this during installation.
12712	TCP	nmsas.server.port.ts.recovery	Default recovery port used by the Transaction service.	Modify the %NnmDataDir%\nmsas\traffic-master\server.propertiesfile (Windows) or \$NnmDataDir/nmsas/traffic-master/server.properties file (Linux).
12713	TCP	nmsas.server.port.ts.status	Default status port used by the Transaction service.	Modify the %NnmDataDir%\nmsas\traffic-master\server.propertiesfile (Windows) or \$NnmDataDir/nmsas/traffic-master/server.properties file (Linux).
12714	TCP	nmsas.server.port.ts.id	Default port used by the Transaction service.	Modify the %NnmDataDir%\nmsas\traffic-master\server.propertiesfile (Windows) or \$NnmDataDir/nmsas/traffic-master/server.properties file (Linux).

Ports Used on the NNM iSPI Performance for Traffic Management Server (Traffic Leaf)

The following table shows the ports the NNM iSPI Performance for Traffic (Traffic Leaf component) uses on the management server. In case of port conflicts, almost all of these port numbers can be changed using the server.properties file located at: %NnmDataDir%/nmsas/traffic-leaf/server.properties.

Ports Used on the NNM iSPI Performance for Traffic Management Server (Traffic Leaf)

Port	Type	Name	Purpose	Change Configuration
5432	TCP	com.hp.ov.nms.postgres.port	This PostgreSQL port is the port the embedded database listens on for this NNMi management server. The port is expected to be the same as that configured for NNMi in the nms-local.properties file.	N/A
11080	TCP	nmsas.server.port.web.http	Default HTTP port - used for Web UI.	Modify the %NnmDataDir%\nmsas\traffic-leaf\server.propertiesfile (Windows) or \$NnmDataDir/nmsas/traffic-leaf/server.properties file (Linux). You can also change this during installation.
11043	TCP	nmsas.server.port.web.https	Default secure HTTPS port (SSL) - used for Web UI.	Modify the %NnmDataDir%\nmsas\traffic-leaf\server.propertiesfile (Windows) or \$NnmDataDir/nmsas/traffic-leaf/server.properties file (Linux). You can also change this during installation.
11083	TCP	nmsas.server.port.naming.rmi	Default port for RMI naming service	Modify the %NnmDataDir%\nmsas\traffic-leaf\server.propertiesfile (Windows) or \$NnmDataDir/nmsas/traffic-leaf/server.properties file (Linux).
11084	TCP	nmsas.server.port.jmx.jmp	Default RMI object port (JRMP invoker)	Modify the %NnmDataDir%\nmsas\traffic-leaf\server.propertiesfile (Windows) or \$NnmDataDir/nmsas/traffic-

Ports Used on the NNM iSPI Performance for Traffic Management Server (Traffic Leaf), continued

Port	Type	Name	Purpose	Change Configuration
				leaf/server.properties file (Linux).
11085	TCP	nmsas.server.port.jmx.rmi	Default RMI pooled invoker port	Modify the %NnmDataDir%\nmsas\traffic-leaf\server.propertiesfile (Windows) or \$NnmDataDir/nmsas/traffic-leaf/server.properties file (Linux).
11086	TCP	nmsas.server.port.invoker.unified	Default RMI remoting server connector port	Modify the %NnmDataDir%\nmsas\traffic-leaf\server.propertiesfile (Windows) or \$NnmDataDir/nmsas/traffic-leaf/server.properties file (Linux).
11087	TCP	nmsas.server.port.hq	Used for un-encrypted Global Network Management traffic.	Modify the %NnmDataDir%\nmsas\traffic-leaf\server.propertiesfile (Windows) or \$NnmDataDir/nmsas/traffic-leaf/server.properties file (Linux).
11089	TCP	nmsas.server.port.remoting.ejb3	Default EJB3 remoting connector port	Modify the %NnmDataDir%\nmsas\traffic-leaf\server.propertiesfile (Windows) or \$NnmDataDir/nmsas/traffic-leaf/server.properties file (Linux).
11092	TCP	nmsas.server.port.hq.ssl	Used for encrypted Global Network Management traffic.	Modify the %NnmDataDir%\nmsas\traffic-leaf\server.propertiesfile (Windows) or \$NnmDataDir/nmsas/traffic-leaf/server.properties file (Linux).
11099	TCP	nmsas.server.port.naming.port	Default bootstrap JNP service port (JNDI provider)	Modify the %NnmDataDir%\nmsas\traffic-leaf\server.propertiesfile (Windows) or \$NnmDataDir/nmsas/traffic-leaf/server.properties file (Linux). You can also change this during installation.

Ports Used on the NNM iSPI Performance for Traffic Management Server (Traffic Leaf), continued

Port	Type	Name	Purpose	Change Configuration
11712	TCP	nmsas.server.port.ts.recovery	Default recovery port used by the Transaction service.	Modify the %NnmDataDir%\nmsas\traffic-leaf\server.propertiesfile (Windows) or \$NnmDataDir/nmsas/traffic-leaf/server.properties file (Linux).
11713	TCP	nmsas.server.port.ts.status	Default status port used by the Transaction service.	Modify the %NnmDataDir%\nmsas\traffic-leaf\server.propertiesfile (Windows) or \$NnmDataDir/nmsas/traffic-leaf/server.properties file (Linux).
11714	TCP	nmsas.server.port.ts.id	Default port used by the Transaction service.	Modify the %NnmDataDir%\nmsas\traffic-leaf\server.propertiesfile (Windows) or \$NnmDataDir/nmsas/traffic-leaf/server.properties file (Linux).

Ports Used for Communication Between the NNM iSPI Performance for Traffic Management Server and Other Systems

The following table shows some of the ports NNM iSPI Performance for Traffic uses to communicate with other systems. If a firewall separates the NNM iSPI Performance for Traffic from these systems, you must open many of these ports in the firewall. The actual set of ports depends on the set of integrations you configured to use with the NNM iSPI Performance for Traffic and how you configured those integrations. If column 4 indicates *Client*, the NNM iSPI Performance for Traffic connects or sends to this port; if column 4 indicates *Server*, the NNM iSPI Performance for Traffic listens on this port.

Ports Used for Communication Between the Management Server and Other Systems

Port	Type	Purpose	Client or Server
Any Free Port	TCP	Avaya Streaming	Server
Any Free Port	TCP	RTCP Server	Server
22	TCP	Cisco/Avaya SSH Communication	Client

Ports Used for Communication Between the Management Server and Other Systems, continued

Port	Type	Purpose	Client or Server
22/23	TCP	Cisco FTP/SFTP Communication	Server
23	TCP	Avaya Survivable Communication	Client
8000 (Configurable)	TCP	.NET Proxy(IPT shipped)	Client
8443	TCP	Cisco AXL Communication	Client

NNM iSPI Performance for QA Ports

The following table shows the ports the NNM iSPI Performance for QA uses on the management server. In case of port conflicts, almost all of these port numbers can be changed using the `server.properties` file located at: `%NmDataDir%/nmsas/qa/server.properties`.

Ports Used on the NNM iSPI Performance for QA Management Server

Port	Type	Name	Purpose	Change Configuration
5432	TCP	<code>com.hp.ov.nms.postgres.port</code>	This PostgreSQL port is the port the embedded database listens on for this NNMi management server. The port is expected to be the same as that configured for NNMi in the <code>nms-local.properties</code> file.	N/A
54040	TCP	<code>nmsas.server.port.web.http</code>	Default HTTP port - used for Web UI.	Modify the <code>%NmDataDir%\nmsas\qa\server.properties</code> file (Windows) or <code>\$NmDataDir/nmsas/qa/server.properties</code> file (Linux). You can also change this during installation.
54041	TCP	<code>nmsas.server.port.remoting.ejb3</code>	Used for invoking remote ejb calls.	Modify the <code>%NmDataDir%\nmsas\qa\server.properties</code> file (Windows) or <code>\$NmDataDir/nmsas/qa/server.properties</code> file (Linux).
54043	TCP	<code>nmsas.server.port.web.https</code>	Default secure HTTPS port (SSL) - used for Web UI.	Modify the <code>%NmDataDir%\nmsas\qa\server.properties</code> file (Windows) or <code>\$NmDataDir/nmsas/qa/server.properties</code> file (Linux). You can also change this during installation.

Ports Used on the NNM iSPI Performance for QA Management Server, continued

Port	Type	Name	Purpose	Change Configuration
54045	TCP	nmsas.server.port.invoker.unified	Used by jboss remoting service.	Modify the %NnmDataDir%\nmsas\qa\server.properties file (Windows) or \$NnmDataDir/nmsas/qa/server.properties file (Linux).
54046	TCP	nmsas.server.port.naming.port	Default bootstrap JNP service port (JNDI provider)	Modify the %NnmDataDir%\nmsas\qa\server.properties file (Windows) or \$NnmDataDir/nmsas/qa/server.properties file (Linux). You can also change this during installation.
54047	TCP	nmsas.server.port.hq	Used for un-encrypted Global Network Management traffic.	Modify the %NnmDataDir%\nmsas\qa\server.properties file (Windows) or \$NnmDataDir/nmsas/qa/server.properties file (Linux).
54048	TCP	nmsas.server.port.jmx.rmi	Default RMI pooled invoker port	Modify the %NnmDataDir%\nmsas\qa\server.properties file (Windows) or \$NnmDataDir/nmsas/qa/server.properties file (Linux).
54049	TCP	nmsas.server.port.naming.rmi	Default port for RMI naming service	Modify the %NnmDataDir%\nmsas\qa\server.properties file (Windows) or \$NnmDataDir/nmsas/qa/server.properties file (Linux).
54084	TCP	nmsas.server.port.jmx.jrmp	Default RMI object port (JRMP invoker)	Modify the

Ports Used on the NNM iSPI Performance for QA Management Server, continued

Port	Type	Name	Purpose	Change Configuration
				%NnmDataDir%\nmsas\qa\server.properties file (Windows) or \$NnmDataDir/nmsas/qa/server.properties file (Linux).
54086	TCP	nmsas.server.port.invoker.unified	Default RMI remoting server connector port	Modify the %NnmDataDir%\nmsas\qa\server.properties file (Windows) or \$NnmDataDir/nmsas/qa/server.properties file (Linux).
54087	TCP	nmsas.server.port.hq	Used for un-encrypted Global Network Management traffic.	Modify the %NnmDataDir%\nmsas\qa\server.properties file (Windows) or \$NnmDataDir/nmsas/qa/server.properties file (Linux).
54088	TCP	nmsas.server.port.hq.ssl	Used for encrypted Global Network Management traffic.	Modify the %NnmDataDir%\nmsas\qa\server.properties file (Windows) or \$NnmDataDir/nmsas/qa/server.properties file (Linux).
54089	TCP	nmsas.server.port.remoting.ejb3	Default EJB3 remoting connector port	Modify the %NnmDataDir%\nmsas\qa\server.properties file (Windows) or \$NnmDataDir/nmsas/qa/server.properties file (Linux).
54092	TCP	nmsas.server.port.hq.ssl	Used for encrypted Global Network Management traffic.	Modify the %NnmDataDir%\nmsas\qa\server.properties file (Windows) or \$NnmDataDir/nmsas/qa/server.properties

Ports Used on the NNM iSPI Performance for QA Management Server, continued

Port	Type	Name	Purpose	Change Configuration
				file (Linux).
54712	TCP	nmsas.server.port.ts.recovery	Default recovery port used by the Transaction service.	Modify the %NnmDataDir%\nmsas\qa\server.properties file (Windows) or \$NnmDataDir/nmsas/qa/server.properties file (Linux).
54713	TCP	nmsas.server.port.ts.status	Default status port used by the Transaction service.	Modify the %NnmDataDir%\nmsas\qa\server.properties file (Windows) or \$NnmDataDir/nmsas/qa/server.properties file (Linux).
54714	TCP	nmsas.server.port.ts.id	Default port used by the Transaction service.	Modify the %NnmDataDir%\nmsas\qa\server.properties file (Windows) or \$NnmDataDir/nmsas/qa/server.properties file (Linux).

NNM iSPI Performance for Metrics and NPS Ports

The following table shows the ports required for NNM iSPI Performance for Metrics and Network Performance Server (NPS). In case of port conflicts, almost all of these port numbers can be changed.

Note: If NNMi and NPS are not coexisting, then the network ports used for the OS network file sharing are also required (NFS services on Linux, Windows File Sharing on Windows).

Required Ports for NNM iSPI Performance for Metrics and NPS

Port	Type	Name	Purpose	Change Configuration
9300	TCP	NPS UI	Default HTTP port - used for Web UI & BI Web Services.	Change using <code>configureWebAccess.ovpl</code> .
9301	TCP	Sybase ASE	Sybase ASE (BI Content Manager Database). Used by processes running on the same server.	Change not supported.
9302	TCP	Sybase IQ Agent	Sybase IQ Agent service. Used by processes running on the same server.	Change not supported.
9303	TCP	Sybase IQ - PerfSPI DB	Sybase IQ database used to store all NPS extensionPack data. Used by processes running on the same server.	Change not supported.
9305	TCP	NPS UI - SSL	Default Secure HTTPS port (SSL) - used for Web UI & BI Web Services.	Change using <code>configureWebAccess.ovpl</code> .
9306	TCP	Database SQL Rewrite Proxy - PerfSPI DB	SQL Rewrite proxy for the Perfspi database - used by BI Server. Used by processes running on the same server.	Change not supported.
9308	TCP	Sybase ASE Backup Server	Sybase ASE backup server for the BI content manager database. Used by processes running on the same server.	Change not supported.

NNM iSPI NET Ports

The following table shows the ports used by the NNM iSPI NET Diagnostics Server. The NNM iSPI NET diagnostic server installs HPE Operations Orchestration (OO). For more information, see the *HPE Operations Orchestration Administrator's Guide*.

Ports Used by the NNM iSPI NET Diagnostics Server

Port	Type	Name	Purpose	Change Configuration
3306	TCP	MySQL database port	Provides access to MySQL database.	Change not supported.
8080	TCP	jetty http port	Default HTTP port - used for Web UI & Web Services.	Post-install modifications not supported.
8443	TCP	jetty SSL/https port	Default HTTPS port - used for Web UI & Web Services.	Post-install modifications not supported.
9004	TCP	OO RAS port	Provides access to OO Remote Action Service.	Change not supported.

NNMi Configuration Issues

This section includes common issues with NNMi configuration and the steps to resolve these Issues.

Incorrect Display of the SNMP Data and MIB Strings

NNMi displays garbled strings from some SNMP traps and other octet string data, such as `sysDescription` and `sysContact`, if an incorrect character set is configured in the `nms-jboss.properties` file.

To resolve this problem, do the following:

1. Edit the following file:
 - *Windows:* %NNM_PROPS%\nms-jboss.properties
 - *Linux:* \$NNM_PROPS/nms-jboss.properties
2. Remove the comment (`#!` characters) from the line that begins as follows:


```
#!com.hp.nnm.sourceEncoding=
```
3. Set the `com.hp.nnm.sourceEncoding` JVM property to a comma-separated list of source encodings that your environment currently supports using the examples shown in the `nms-jboss.properties` file. These examples show combinations of the Shift_JIS, EUC_JP, UTF-8, and ISO-8859-1 character sets.
4. Save your changes.
5. Restart the NNMi management server:
 - Run the `ovstop` command on the NNMi management server
 - Run the `ovstart` command on the NNMi management server
6. To test your changes, resend the suspect trap to NNMi and make sure the garbled display problem no longer occurs.

If the garbled text involves binary data or data that cannot be interpreted for any reason, do the following to configure NNMi to display the strings in hexadecimal format:

- a. Open the following file:
 - Windows:* %NNMDATADIR%\shared\nnm\conf\nnmvbnosrcenc.conf
 - Linux:* \$NNMDATADIR/shared/nnm/conf/nnmvbnosrcenc.conf
- b. Add the trap OID - varbind OID value combinations that NNMi displays in a garbled format. Also add the combinations from any varbind values you do not want NNMi to decode, such as binary data. Use the examples shown in the `nnmvbnosrcenc.conf` file as templates to configure your combinations. This tells NNMi to display the Custom Incident Attribute values in the Incident form using a hexadecimal value.
- c. Save your changes.
- d. Restart the NNMi management server:
 - Run the `ovstop` command on the NNMi management server.
 - Run the `ovstart` command on the NNMi management server.
- e. Test your changes to make sure these changes result in a hexadecimal display of the formerly garbled strings.

ESXi Servers Appear as Devices with no SNMP on NNMi Maps

To address this, make sure that the SNMPM agent is installed and enabled on the ESXi server..

Problem with the Node Group Map

A node group map may show an error message when your NNMi environment includes the following:

- An Oracle database.
- A top-level NNMi node group with multiple child node groups
- A child node groups contain 1000 or more members

To address this, do one of the following:

- Limit the child node groups to less than 1000 members
- Do not select **Nodes to Node Groups** or **Node Groups to Node Groups** in the **Node Group Map Settings->Connectivity->Node Group Connectivity** section for these node groups.

Accidental Removal of the Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy Files Library

If you accidentally removed the Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy Files library, and need to enable NNMi to use the AES-192, AES-256, and TripleDES privacy protocols for SNMPv3 communication, follow these steps:

1. Download the Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy Files library from the Oracle Technology Network web site for Java developers.
2. Uncompress the download package, and then copy both JAR files (`local_policy.jar` and `US_export_policy.jar`) to the following location:
 - *Windows:* %NmInstallDir%\nonOV\jdk\nnm\jre\lib\security
 - *Linux:* \$NmInstallDir/nonOV/jdk/nnm/jre/lib/security
3. Restart the NNMi management server:
 - a. Run the **ovstop** command on the NNMi management server.
 - b. Run the **ovstart** command on the NNMi management server.

Send Documentation Feedback

If you have comments about this document, you can [contact the documentation team](#) by email. If an email client is configured on this system, click the link above and an email window opens with the following information in the subject line:

Feedback on Deployment Reference (Network Node Manager i Software 10.30)

Just add your feedback to the email and click send.

If no email client is available, copy the information above to a new message in a web mail client, and send your feedback to network-management-doc-feedback@hpe.com.

We appreciate your feedback!